

**DEVELOPMENT OF COMPUTER ALGORITHM FOR CONSTRUCTION  
OF APTITUDE TEST BATTERY FOR COMPUTER PROGRAMMERS**

**PROJECT REPORT**

**D. Dutta Roy**



**Psychology Research Unit**  
**INDIAN STATISTICAL INSTITUTE**  
**203, B.T. ROAD**  
**KOLKATA - 700 108**

**1998-2002**

---

## Contents

---

	Pages
ABSTRACT	5-6
ACKNOWLEDGEMENT	7
<b>CHAPTER 1</b>	
PERSPECTIVES, DEFINITION AND OBJECTIVES	
1.1 Usefulness of computer aided testing	8
1.2 Definition of Aptitude	9-10
1.3 Aptitude tests	10-12
1.4 Computer programmer aptitude tests	12-15
1.5 Objectives	15
<b>CHAPTER 2</b>	
METHODS	
2.1 Sample	18
2.2 Instruments	19
2.3 Procedure	20
<b>CHAPTER 3</b>	
RESULTS	
<b>Job analysis</b>	26-35
Task importance for all groups	26
Task importance across occupations	27
Clustering occupations	27
Softwares used	27
Working conditions	29
Team assignment	29
Social interaction	29
Occupational disorders	30
<b>Computer programming task taxonomy</b>	37-41
Factor analysis	38
Factor scores	39
Differences across occupations	39
<b>Relative importance of aptitudes</b>	42-50
Reliability of judgements	44
Relative importance of aptitudes	44
Rank order correlations	45
Cluster analysis	45
Cluster wise aptitude importanace	46
<b>Relationship between task and aptitude importance</b>	51-58
Program writing and testing	53
Human relations	53
Data analysis	53
User satisfaction	54
<b>CHAPTER 4</b>	
DESCRIPTION OF COMPUTER AIDED APTITUDE TESTS	59-66
Absurdity	60
Anagram	60

Vocabulary	61
Number series	61
Inference	61
Numerical reasoning	62
Coding and Decoding	62
Memory for commands	62
Short term memory test	63
Comprehension	63
Creativity	63
<b>CHAPTER 5</b>	
SUMMARY FINDINGS	67-68
FURTHER RESEARCHES	69
REFERENCES	70-73
APPENDICES	
Appendix A : Algorithms of test battery	76-180
Appendix B : Questionnaire	181-184

**TABLES**

SL	Title	Page
Table 2.1	Sample frame	22
Table 2.2	Distribution of age across occupations	23
Table 2.3	Distribution of programming experience across occupations	23
Table 2.4	Distribution of academic qualifications across occupations	23
Table 2.4	Differences in observed frequencies across programming tasks	24
Table 3.1	Medians and Quartile deviations of computer programming tasks (n=201)	30
Table 3.2	Occupation wise task importance in median values	31
Table 3.3	Euclidean distance matrix	32
Table 3.4	Softwares used	33
Table 3.5	Working conditions	33
Table 3.6	Team assignment pattern across occupations	34
Table 3.7	Distribution of interaction with other programmers	34
Table 3.8	Distribution of Occupational disorders	35
Table 4.1	Correlation matrix of programming tasks (N=201)	39
Table 4.2	Factor analysis of programming tasks	40
Table 4.3	One way ANOVA in programming task factors across different occupations	41
Table 5.1	Relative importance of Aptitudes across Occupations	46
Table 5.2	Spearman Rank-order correlation coefficients among different occupational groups in their judgement on relative importance of computer programming aptitudes	48
Table 5.3	Euclidean distances among different occupational groups in their judgement on relative importance of computer programming aptitudes	48
Table 5.4	Aptitude importance across three clusters in median values	50
Table 6.1	Correlation coefficients between program writing and testing and aptitudes across occupations	54
Table 6.2	Correlation coefficients between importance on Human relations and aptitudes across occupations	55
Table 6.3	Correlation coefficients between importance on Data analysis and aptitudes across occupations	56
Table 6.4	Correlation coefficients between importance on User satisfaction and aptitudes across occupations	57
Table 6.5	No of aptitudes significantly related to programming task factors (n=201)	58
Table 6.6	No of aptitudes significantly related to programming task factors among six occupations	58

**FIGURES**

Sl	Title	Page
Figure 1.1	Hypothetical model	17
Figure 2.1	Correspondence between task and rating categories	25
Figure 3.1	Dendogram based on occupation wise median values across computer programming tasks	32
Figure 5.1	Dendogram of occupations based upon relative importance of aptitudes	49
Figure 6.1	Flow chart for absurdity, anagram, coding and decoding, inference, comprehension, vocabulary aptitude test battery	64
Figure 6.2	Flow chart for number series and arithmetical reasoning tests	65
Figure 6.3	Flow chart for short term memory tests	66

**ABSTRACT**

Prior studies on aptitudes were limited to aptitude test development for selection, vocational counseling and guidance. In test development, attention was paid to the intuition of test developers resulting gap of knowledge about relative importance of aptitudes for the job occupant. Limitation in understanding cognitive appraisal of aptitudes causes serious problem in vocational counseling and guidance. Before construction of computer aided aptitude test battery, current project attempted to explore relative importance of aptitudes across different occupations of computer programmers and relationship between aptitude and task importance. Data were collected from 201 experienced computer programmers from six occupations (manufacturing, financial, research and development, administration, research and education, computer training institutes) through checklist. Programmers were asked to rate to what extent the listed programming task and aptitudes were important for successful performance in their occupations. Some multivariate statistics were used for data analysis. Correspondence analysis was made for item analysis of checklist. Factor analysis revealed latent relationship among programming tasks and it extracted four task factors – program writing and testing, Human relations, Data analysis and User satisfaction. . Using ANOVA, it was noted that besides data analysis, other task factors were not significantly different across occupations. Disregarding occupational differences, five specific aptitudes namely comprehension,

inference, expressional fluency, associational fluency and alternative usage were found relatively more important. The last three aptitudes were not found in earlier study on computer programming aptitude. Cluster analysis classified six occupations into two clusters in terms of task and into three clusters in terms of aptitude importance. Results noted correlation coefficients between task and aptitude importance across occupations. In considering relative importance of aptitudes, a computer aided test battery was developed. The test battery consists of twelve general aptitudes as absurdity, anagram, vocabulary, number series, inference, arithmetical reasoning, coding and decoding, memory for commands, short term memory, comprehension, and creativity using three different programming languages – QBASIC, JAVA, and C

## **ACKNOWLEDGEMENT**

I express my gratitude to Professor K.B. Sinha, the Director of ISI, Professor B.L.S. Prakasa Rao, the Ex-Director of ISI, Professor S.R. Chakravarty, Professor-in-charge of the Social Sciences Division, Professor B.N. Bhattacharya, then Professor-in-charge of the Social Sciences Division, Dr. Anjali Ghosh, the Head of the Psychology Research unit for administrative support and encouragement.

I am indebted to four project assistants-Mrs. Ishani Banerjee, Mrs. Rama Mullick, Mr. Subhabrata Mallick and Nandini Biswas for their assistance across different phases of this project. Mrs. Ishani Banerjee assisted me in developing questionnaire and checklists. Mrs. Rama Mallick assisted me in collection of data through the questionnaire and checklists. Mr. Subhabrata Mallick assisted me in developing program on multimedia based memory test. Miss Nandini Biswas joined at the last moment and assisted me in developing items for various aptitude tests.

I am grateful to the programmers of different industries, banks, training institutes, administrative departments, research institutes and universities for spending their valuable time in filling up the questionnaires.

I am grateful to Professor Amitava Chatterjee, Ex-Head of the Department of HSS, IIT,, Kharagpur, Professor Manjula Mukerjee, Ex-Head of the Psychology Research Unit of ISI, Dr. Debashish Sengupta, Associate Professor of Applied Statistics Unit for their expertise and valuable suggestions in selection of project assistant. I am thankful to my colleagues and staffs of the Psychology Research unit and other units and divisions of ISI for their consistent support and encouragement in this project.



### **Perspectives, Definitions and Objectives**

In selection and vocational guidance based on aptitudes, computer aided aptitude assessment is more useful than manual assessment because of the followings(Cronbach,1970):

- a) **Excellence of standardization:** the computer delivers precisely the test items called for by the program, with no unintended variation in timing, tone of voice or prompting.
- b) **Sequential administration:** The test items need not be the same for everyone. As the person succeeds, the computer tests his limits with harder task ; as he does poorly , the computer offers tasks at a lower level. Greatest time can be given to testing aspects of performance where weakness or exceptional talent appears. Something like this is done by the psychologist testing one individual at a time , but a computer method can do it more systematically.
- c) **Precision of timing:** While speed is not generally emphasized in current testing, precise timing of individual responses may be valuable, specially in investigating learned skills and in evaluations of brain damage.
- d) **Release of human testers for other duties:** In so far as testing a routine function it is worst to use a well-educated tester
- e) **Patience:** The computer's patience is inexhaustible. The human tester may be bored due to repeated failure of the testing but the computer can wait until the subject makes a response.

f) **Control of bias:** The human tester may be subjectively due biased to the testee's shiny face, curly hair and starched dress (Hawthorne effect). But the computer does not consider this phenomena; rather takes into account testee's response.

Therefore, the major objective of this project was to develop computer algorithm for construction of General Aptitude Test Battery.

### **Aptitude**

Assessment of aptitude is important for selection, training, counseling and vocational guidance. An aptitude is a combination of characteristics indicative of an individual's capacity to acquire (with training) some specific knowledge, skill or set of organized responses related to specific occupation. Warren in the ,Dictionary of Psychology, defined aptitude as a "condition or set of characteristics regarded as symptomatic of an individual's ability to acquire with training some knowledge, skill, or set of responses such as the ability to speak a language, to produce music, etc.

Aptitude is distinguished from ability and achievement. It should be noted that aptitude is differentiated from skill and proficiency. *Skill* means the ability to perform a given act with ease and precision . *Proficiency* has much the same meaning, except that it is more comprehensive; for it includes not only skills in certain types of motor and manual activities, but also in other types of activities as shown by the extent of one's competence in language, history, economics, mathematics. We may speak of one's degree of proficiency in any type of performance. On the other hand, when we speak of an individual's *aptitude* for a given type of activity ,we mean the capacity to acquire proficiency under

appropriate conditions; that is, his potentialities at present, as revealed by his performance on selected tests that have predictive value. There is no widely accepted definition of the difference between ability and aptitude. Most people would agree that to some extent the two terms refer to the same thing: aptitude referring to specific ability, and ability referring to general aptitude. We could probably view ability as underlying aptitude, and aptitude as being more job related than ability. For instance a computer programmer might score highly on a verbal ability test and highly on a programmer aptitude test but not the other way around.

### **Aptitude tests**

An aptitude test is one designed to measure a person's potential ability in an activity of a specialized kind and within a restricted range (Freeman, 1968 ). Aptitude tests are to be distinguished from those of general intelligence and from tests of skill or proficiency acquired *after* training or experience. They should be distinguished ,too, from educational achievement tests , which are designed to measure an individual's *quantity* and quality of learning in a specified subject of study after a period of instruction.

The development of aptitude tests received impetus in World War I in considering the army's need to select men for tasks requiring specific skills; the desire ,in vocational guidance and personnel assessment Aptitude tests have been developed to predict educability and performance in music and drawing ,in mechanical and clerical occupations, in engineering, in medicine and law ,and in other areas as well. Others in this category are intended to evaluate aptitudes for the

study of specific types of subject matter, such as science, foreign languages and mathematics.. Aptitudes are of multifactorial in nature covering (i) mental abilities, (ii) mechanical and related abilities (iii) psychomotor abilities (iv) visual skills and (v) a class of other specialized aptitudes. Mental abilities include verbal comprehension, word fluency, memory, inductive reasoning, number facility, speed of perception, spatial visualization (Thurstone, 1948). Typical mental ability tests are :

Otis self-administering tests of mental ability, Wonderlic personnel test, the adaptability test, Purdue Non-language Personnel test and the SRA Primary mental abilities tests. Mechanical ability includes two things as mechanical comprehension and understanding the mechanical principles. Some of the widely used tests to assess mechanical aptitudes are Bennett Test of Mechanical comprehension, Flanagan aptitude classification, Minnesota mechanical assembly test, Minnesota spatial relations test and Purdue mechanical performance test. Psychomotor abilities include a series of ability factors as control precision, multilimb coordination, response orientation, reaction time, speed of arm movement, rate control, manual dexterity, finger dexterity, arm-hand steadiness, wrist finger speed, aiming (Fleishman, 1954; Fleishman and Hemphill, 1956; and Fleishman and Ornstein, 1960). Typical psychomotor tests are O'connor Finger dexterity test, Hayes pegboard, Stromberg dexterity test. Extensive research on the visual skills of people on various jobs in industry and business has indicated certain aspects of visual performance as being most important in classifying and placing employees according to differences in visual characteristics, these aspects being, keenness of vision(visual acuity),discrimination of differences in distance(depth perception or

stereopsis), discrimination of differences in color and postural characteristics of the eyes. Two of the well-known vision tests in use are Snellen letter chart and American Medical Association notation. Besides vision-testing devices such as the Ortho-Rater, the Sight-Screener, the Telebinocular, and the Vision Tester are extensively used for selection and placement purposes. Aside from the types of aptitude tests discussed above, there are many others that deal with other special aptitudes or attributes. Among such tests are those of clerical aptitude, reading speed and comprehension, vocabulary, and perceptual speed. Some such tests are Seashore-Benett Stenographic Proficiency test, S.R.A Typing skills, S.R.A Dictation Skills, General Clerical Test, Minnesota Clerical test, Purdue Reading Test for Manufacturing Supervisors, Purdue Manufacturing Supervisors' Word-Meaning Test and Purdue Creativity Test.

Prior studies have been concerned with aptitudes of system engineers (Toshima,1993), teachers(Kurkreti,1993 and 1985,Bisht,1982;Raina,1981),science students(Sharma and Shukla, 1992; Mehta, 1982; Chatterjee and Mukerjee,1978; Agarwal,1977; Dev, 1968), Manufacturing technical personnel (Mohan and Brar, 1988), office workers (Naik,1973), engineers (Deb,1973), special class railway apprentices (Avtar,1965) and paper folders(Snaw,1970).Comparatively least attention has been paid to determine aptitudes of computer programmers.

### **Computer programmer aptitude tests**

#### *1. Computer programmer aptitude tests by SRA :*

Cronbach (1970) noted that Science research associates (SRA) of Chicago developed a test to measure the aptitudes of computer programmers in 1964. The

test battery includes verbal meaning, number series, numerical ability, reasoning, ingenuity, and diagramming. Using supervisors' ratings, it was noted that validity coefficients of the test battery ranged from 0.30 to 0.50.

2. *The Berger Aptitude for Programming Test:*

Form D (B-APT D) test was designed by Psychometrics Inc for measuring the ability of a person to learn the skills necessary to succeed in programming training for the mainframe world. The B-APT is designed so that talented examinees with no programming background can understand the test's programming language and use to solve the problems. It is a work sample test in which the examinee writes coded instructions to a "computer" in a logical sequence to carry out program specifications. The examinees must apply the rules of coding, looping, incrementing and branching as they write short programs in the test booklet. They must also be able to demonstrate "adaptive flexibility" by applying the language rules to progressively more complex problem requirements. The B-APT is composed of 30 questions which require the examinee to write short program segments in the B-APT's own simplified programming language. The total time to administer the test is 1 hour and 15 minutes. The B-APT is also available in braille and Enlarged versions for the visually impaired.

3. *Berger Aptitude for Programming Test Advanced Form (B-APT AF) :*

It is an advanced form of the B-APT . This test is meant for those who have prior programming experience such as Computer Science degree holders. Candidates should have had some computer science courses or similar study or experience. The test is also commonly used to select from the COBOL programming group those to

be trained in newer languages and technologies such as client server, or C language. The B-APT AF is composed of 25 questions which require the examinee to write short program segments in the B-APT AF's own simplified programming language. The total time to administer the test is 2 hours.

#### *4. The Walden Programmer Analyst Aptitude Test*

Walden Personal Testing and Training Services have come up with tests that help in assessing the true skills of potential programmers. These tests are designed to be used in conjunction with in-depth interviews. The test, namely, Programming Analyst Aptitude Test consisted of six questions developed around the following six high-level abilities:

- Follow a complete procedure involving multiple transactions.
- Create a routine procedure that solves a problem through the structuring of tasks. Problems involving the first two abilities are developed around common banking procedures,
- Interpret detailed specifications, follow rules, and incorporate symbols to solve problems. Problems testing this ability was developed around the general structure of computer machine language instructions (concepts of operation codes, data length, address, and so on).
- Analyze and execute a new instruction set, perform table look-up, and manipulate data to generate a solution. , Problems testing this ability forced the individual to perform complicated table look-up procedures and then manipulate the data found to solve progressively difficult mathematical expressions.

- Analyze a complex business situation and search through voluminous data to understand and identify core requirements. The problem testing this ability required detailing a complex business procedure in narrative form. It was modeled after a complicated billing system for a fictitious company
- Create a symbolic program to solve a business-related problem and then identify errors based on generated code. Problems testing this ability had the candidate write a symbolic program based on an instruction set the individual learns from specifications provided at the beginning of the problem

The predictive validity of PAAT is demonstrated by the fact that there exists an empirical relationship ( $r=0.457$ ) which is significant to the 0.01 level between high scores on the PAAT and programmer performance on the job. For the content validation of the PAAT, 14 key functions were identified for the job position through complete job analysis. Given the demonstrated relationship between the abilities and traits required to perform the 14 essential functions of the programmer analyst job and those measured by the PAAT (69.3% of the intended domain as measured by the test), the test represented a content-valid evaluation device for that job.

### **Objectives:**

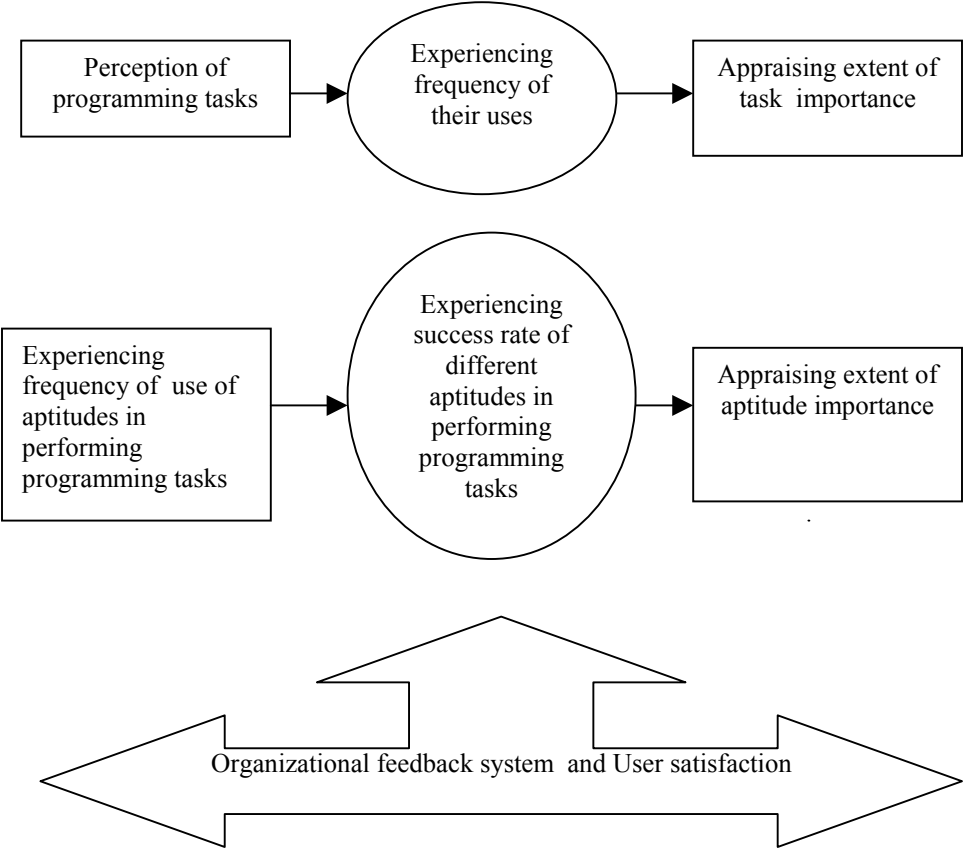
Aptitude is job specific. Computer programming job is very complex, dynamic and multi factor in nature as it includes wide range of activities- Again the activities vary across different occupations. Therefore, current study attempted to explore following objectives:



1. To explore job description (analysis of programming tasks, working conditions, required qualifications and physical characteristics).of computer programmers .  
In assessing task importance a significant correlations among the computer programming tasks was noted. This led to think some latent phenomena in the set of tasks.
2. To explore factor analysis of computer programming tasks.
3. To determine significant differences in computer programming task factors across occupations.
4. To determine relative importance of aptitudes across occupations
5. To determine correlations between computer programming task factors and the aptitudes across occupations
6. To develop computer algorithm for construction of general aptitude test battery for computer programmers.

7.

Figure 1.1  
Hypothetical model



## METHODS

### *Sample:*

Sampling was done in two stages - a) selection of sample organizations and b) selection of sample programmers. For selection of sample organizations, a list of industries and service organizations where in maximum utilization of computer programming was going on were collected . For this purpose, computer-training institutes in Calcutta and Calcutta industrial fare were visited. Newspapers, business magazines, and internets were surveyed to collect the names of the organizations. Names of total 65 organizations (Table 2.1) were collected in this way. 279 names of programmers with more than 3 years experience in computer programming in respective organizations were collected from these 65 organizations. Finally, data were collected from 201 computer programmers . Of them, 34 (17%) from manufacturing, 40 (20%) from financial organizations, 43 (21%) from scientific organizations, 30 (15%) from software development industries, 33 (16%) from computer training institutes, and 21 (10%) were from other departments (Table 2.1) . Their mean age was 35.60 with SD 8.83 years. Table 2.2 shows significant ( $F(5,194)=7.95, p<0.00001$ ) differences in age across the occupations. Programmers in manufacturing organizations (Mean age = 38.91, SD=10.01) were elder and those in R & D (Mean age =30.64, SD=5.39) were younger than the others. Table 2.3 represents significant differences ( $F(5,173) = 4.60, p< .0006$ ) in experiences across occupations. Programmers in manufacturing organizations (Mean years of experience = 10.00, SD= 4.89) were more experienced and programmers in R & D (Mean years of experience=5.87, SD=4.05) were less experienced in comparison

with other programmers. Table 2.4 shows that 184 (95%) programmers reported their educational qualifications. Of them, 182 (99%) programmers were above higher secondary education. There were more number of persons with graduation in Manufacturing(76%) and administration (0.67%) where as more with post graduation qualification in computer training institutes (56%). Here post graduation includes post graduate diploma in computer applications. Relating to nature of project ,49 (35%) programmers reported that their project was repetitive in nature whereas 95 (65%) respondents reported that their project was time-bound in nature.

*Instrument :*

*1. Task analysis :*

On the basis of literature survey (Jain,1987; Balgurusamy,1991; Roy, 1996), different discussions in the internet on computer programming, and open ended interview with 10 experienced (more than 5 years) computer programmers, a checklist with 14 items was prepared to determine relative importance of computer programming tasks. Sample respondents were asked to judge to what extent each item of the checklist was suitable for their job using six point rating scale. The rating categories are:

0=Inapplicable : does not occur in the job

1=May occasionally be a minor part of the job

2=A small part of the job

3=A moderately significant part of the job

4=A substantial part of the job

5=A very significant part of the job

Significant differences (Chi-square = 976.74 df=65 p=0.00001) in observed frequency across six rating categories suggest discriminating validity of the items

(Table 2.5)..Figure 2.1 shows extent of distance between task and rating categories based on correspondence analysis. It was noted in the figure that three rating categories namely moderately, substantially and significantly were located at the left hand side and the other categories were located at the right hand side of the graph. Cronbach's alpha shows high internal consistency ( $\alpha = 0.74$ ) among the items.

## *2. Aptitude analysis :*

A checklist was prepared with 40 different specific aptitudes to explore relative importance of computer programming aptitudes. Samples were asked to judge how far the listed aptitudes are required for satisfactory performance on their job using a 4-point rating scale ranged from not required (score -1), medium to a low degree in some element of job (score - 2), above average in numerous elements (score - 3) and very high degree in all elements of job (score - 4). The list of aptitudes were collected from different literatures on psychological testing (Freeman,1962; Anastasi, ,Cronbach,1967; Tiffin and McCormick,1971). Split half reliability test shows very high internal consistency ( $r_{tt} = 0.88$ ) among the items of checklist. Mean of the first half of items was 54.06 with SD 8.37 and that of the second half was 54.95 with SD 8.98.

## *Procedure*

Initially, a formal letter of request for collection of data was submitted to the higher authority of various organizations. Then the list of programmers with their years of working experiences was collected. Programmers with more than 3 years of experiences were identified as samples of the study . In sampling attention was paid to the more experienced persons when there were many programmers with more than 3 years of experinces. Questionnaire was distributed individually to 279

programmers with assurance that individual responses would not be disclosed anywhere. Of them 201 programmers returned their responses.

Table 2.1

Sample frame

<i>Occupations</i>	<i>Names of the organizations</i>	<i>N</i>	<i>Occupations</i>	<i>Names of the organizations</i>	<i>N</i>	
<b>Manufacturing</b>	Texmaco	8	<b>Administration</b>			
	Balmer Lawrie	1		Bengal chamber of commerce	1	
	ECL	2		Andrew and Yule	1	
	DRS technology	3		ISI	1	
	DPS	1		Railways	7	
	Calender exhibition	1		Shipping corporation	3	
	SAIL	8		CSTC	3	
	Polar fans	2		Kothari medical centre	2	
	Tide water and oil company	2		JMD medicare	1	
<b>Financial</b>	Andrew and Yule	6	Birla heart research centre	1		
	Peerless	13	Suraksha diagnostic centre	1		
	National insurance	7	<b>Research and Education</b>			
	UBI	9		Department of post	2	
	Manufacturing development board	2		INSDOC	1	
	UCO	1		CALIBNET	1	
	General finance	1		CCLN	1	
	LIC	2		IIT., Kgp.	8	
	Reserve bank	4		ISRO	2	
	Cent EDB	1		ISI	1	
	<b>R &amp; D</b>	NSS		2	Telecom computer centre	1
		CESC		8	Aajkal	1
		WBSEB		4	ABP	3
DRS		4		Calcutta telephones	6	
WAPCOS		1		Calcutta university	2	
Irrigation and water ways department		2	Information business system	1		
National informatic centre		7	<b>Computer training institutes</b>			
ISI		2		Calcutta university computer centre	9	
ARB		2		TULEC	6	
Indicon management		3		NIIT	4	
Lee and nee softwares		1		Brainware	4	
ADA software		1		APTECH	1	
CTS		1		CMC	1	
IMRB		4		MCKV institute of engineering	5	
Jadavpur university		1		ENKON computers	1	
				Information software private limited	1	
				First computers	1	

Table 2.2  
Distribution of age across occupations

Organizations	Mean	SD	Valid N
Manufacturing(n=34)	38.91	10.01	34
Financial (n=40)	38.85	6.42	40
R & D (n=43)	30.64	5.39	43
Administration (n=21)	38.48	9.07	21
Research and educational institutes (n=30)	37.03	9.30	30
Computer training institutes (n=33)	31.42	9.03	33
Total	35.60	8.83	201

$F(5,194)=7.95, p<0.00001$

Table 2.3  
Distribution of programming experience across occupations

	Mean	SD	Valid N
Manufacturing(n=34)	10.00	4.89	31
Financial (n=40)	10.00	5.27	39
R & D (n=43)	5.87	4.05	39
Administration (n=21)	11.11	7.49	19
Research and educational institutes (n=30)	7.08	3.87	25
Computer training institutes (n=33)	6.96	7.02	26
Total	8.37	5.64	179

$F(5,173)=4.60, p<0.0006$

Table 2.4  
Distribution of academic qualifications across occupations

Occupations	Total	H.S	Graduation	Post graduation
Manufacturing(n=34)	34	0	26	8
Financial (n=40)	40	0	16	14
R & D (n=43)	42	2	23	17
Administration (n=21)	21	0	14	7
Research and educational institutes (n=30)	30	0	16	14
Computer training institutes (n=33)	27	0	12	15
Total	194	2	107	75

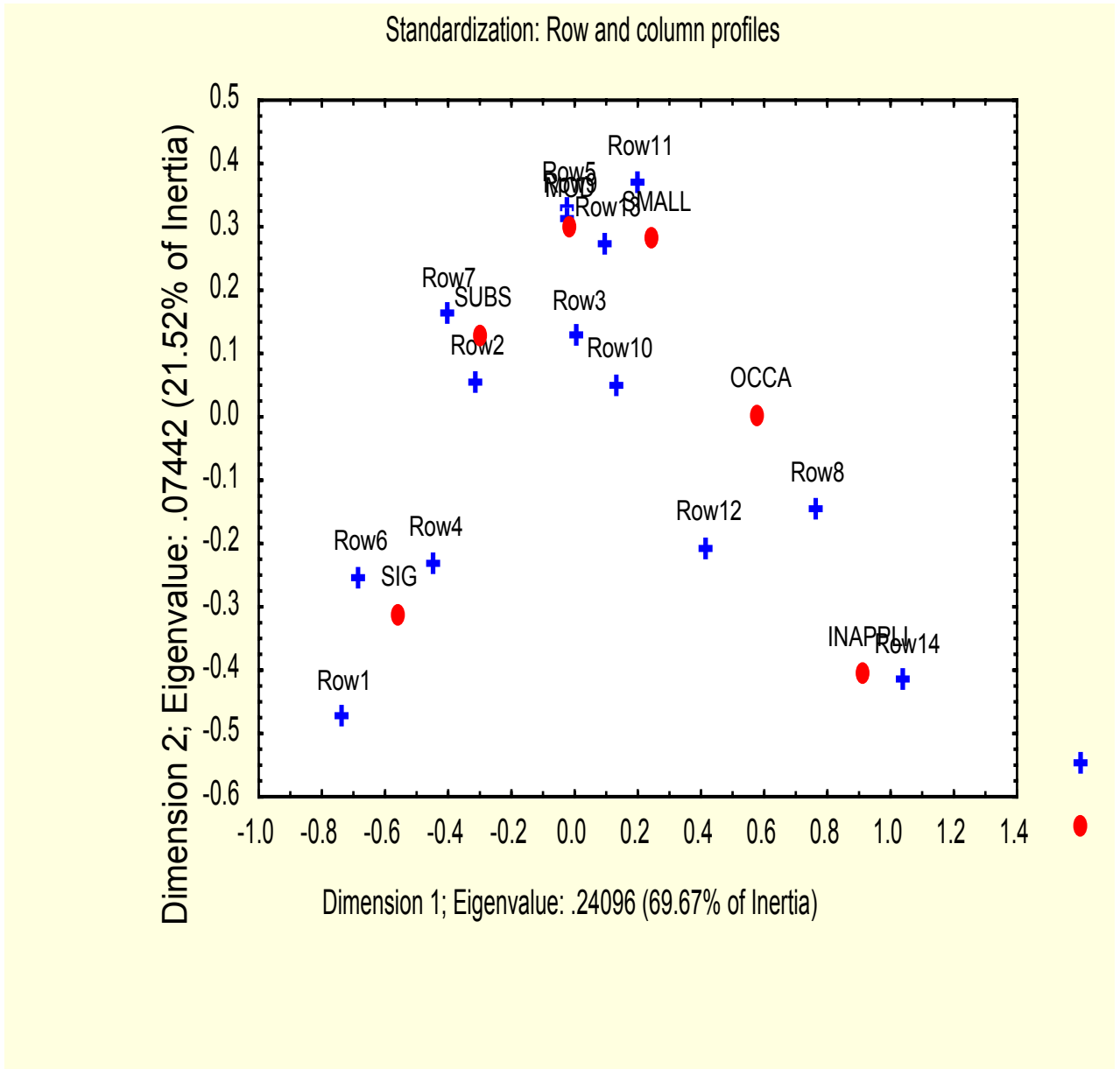
Chi square (15) = 16.37, NS.



Table 2.5  
Differences in observed frequencies across programming tasks

	Inapplic able	Occasio nally	Small part	Moderatel y part	Substatial part	Significa nt part
1. Comprehending users' problem	3	1	11	11	46	122
2. Converting specifications into detailed instruction	13	9	18	39	65	57
3. Drawing Program Flowchart	14	29	36	43	30	49
4. Coding into Computer languages	12	9	17	30	43	90
5. Preparing instruction for operator	13	13	54	36	52	33
6. Testing and correcting program	4	3	6	31	56	101
7. Modifying program	3	9	20	50	58	61
8. Entering data	59	48	33	29	16	16
9. Documenting program	11	24	33	53	46	34
10. Supervising juniors	36	10	35	41	43	36
11. Assisting other programmers	19	29	38	55	42	18
12. Designing computer configuration for users	50	38	28	26	30	39
13. Writing simple program to retrieve data	11	37	38	41	42	32
14. Collaborating with computer manufacturer	88	43	30	19	10	11
Total	336	309	397	504	579	699

Chi-square = 976.74 df=65 p=0.00001



**Fig. 2.1**  
Correspondence between task (Row) and rating categories

Row1: Comprehending, Row 2: Converting, Row 3: Drawing, Row 4: Coding, Row 5: Instructing, Row 6: Testing, Row 7: Modifying, Row 8: Entering, Row 9: Documenting, Row 10: Supervising, Row 11: Assisting, Row 12: Designing, Row 13: Writing, Row 14: Collaborating

## RESULTS

### **Job Analysis**

#### **A: Theoretical considerations**

Job analysis refers to the process of obtaining information related to job description and job specification. Job description includes job summary, work performed, equipment used and working conditions. Job specification includes specific potentialities to perform the work effectively. Job analysis is important for personnel recruitment, training and personnel development, wage and salary administration, performance measurement and rating, job design , organizational planning, manpower planning and control, vocational counseling and guidance (Tiffin and McCormick,1971). Common methods for job analysis are observation, interview, checklist, work diaries and moving pictures.

In analysis of computer programming job, attempt was made to identify relative importance of computer programming tasks ,working conditions, specialized qualifications and physical characteristics required for success in computer programming jobs through observation and checklist..

#### **B. Results**

##### *Task importance disregarding occupational difference*

Table 3.1 shows that computer programmers gave relatively more importance on comprehending users' problems (Mdn=5.00, QD=0.50) , testing and correcting programs (Mdn=5.00, QD=0.50), converting specifications into detailed instructions (Mdn=4.00,QD=1.00), coding into computer languages (Mdn=4.00, QD=1.00) and modifying program (Mdn=4.00,QD=1.00) than drawing program

flow-chart (Mdn=3.00, QD=1.00), preparing instruction for operator (Mdn=3.00,QD=1.00), documenting program (Mdn=3.00, QD=1.00), supervising juniors (Mdn=3.00,QD=1.00), assisting other programmers (Mdn=3,QD=1.25), writing simple program to retrieve data Mdn=3,QD=1.00) designing computer configuration for user (Mdn=2.00,QD=1.50),data entry(Mdn=1,QD=1.50) and collaboration with computer manufacturer (Mdn=1.00,QD=1.00).

#### *Task importance across occupations*

Table 3.2 shows occupation wise median values across each programming tasks. There was not much variations in task importance across occupations.

#### *Clustering the occupaitons*

Using median values of Table 3.2, single linkage method of cluster analysis (Fig 3.1) noted two clusters. Table 3.3 represents Euclidean distance matrix. Cluster 1 included 4 occupations - manufacturing, research and development, administration and research and educational institutes. Cluster 2 included finance and computer training institutes. However, both organizations in cluster 2 were widely distant from each other.

#### *Softwares used*

Related to software use, significant differences Chi square ( 20) =55.36,  $p < 0.00$  were noted across occupations. In programming assignment, results (Table 3.4) show that 98% programmers mostly used different language based programs. Findings are in consistent with the results of task importance (Table 3.1). Programmers in financial , R&D organizations and administration reported use of programming languages. Besides, language based program, they (n=177) used softwares for data analysis. Among them mostly used DBM (32%), object oriented softwares (30%), software packages or tools (29%) . DBM included DBASE,

FOXPRO, JAVA, DHTML were few of object oriented softwares. MINITAB, SPSS, BMDP, STATISTICA were the software packages or tools.

#### *Working Conditions*

There was no significant difference (chi-square (12) = 14.71, NS) in working conditions across occupations ( Table 3.5) . Results show that 82% (n=66) of cases worked in cool environment. 10% noted (n = 8) noise in their work place.

#### *Team assignment across occupations*

Results noted significant differences (Chi square ( 6) = 30.56,  $p < 0.0001$ ) in team assignment across occupations. Programmers in R & D organizations preferred alone assignment more than other groups ( Table 3.6) .

#### *Social interaction*

Social interaction significantly (Chi square (15) =30.26,  $p < 0.01$ ) varied across occupations depending upon preference for team work ( Table 3.7) . Team work is useful for programming as one total program requires various subprograms or subroutines. It was found that most programmers preferred team work (60%) to alone assignment (40%). Those who were engaged in writing total program related to research problem, preferred alone assignment more. Results show that programmers in R & D organizations (44%) preferred alone assignment than other groups. Whereas programmers in computer training institutes (61%) preferred team work than other groups.

*Occupational disorders*

Table 3.8 shows no significant difference (Chi square (20) =28.32, NS) between occupational differences and occupational disorder. Most of them suffered from spondilosis (37%) and sleep disorders (27%).

Table 3.1

Medians and Quartile deviations of computer programming tasks (n=201)

Programming task characteristics	Medians	QD
1. Comprehending users' problem	5.00	0.50
2. Converting specifications into detailed instruction	4.00	1.00
4. Coding into Computer languages	4.00	1.00
6. Testing and correcting program	4.00	0.50
7. Modifying program	4.00	1.00
10. Supervising juniors	3.00	1.00
11. Assisting other programmers	3.00	1.25
13. Writing simple program to retrieve data	3.00	1.00
3. Drawing Program Flowchart	3.00	1.00
5. Preparing instruction for operator	3.00	1.00
9. Documenting program	3.00	1.00
12. Designing computer configuration for users	2.00	1.50
14. Collaborating with computer manufacturer	1.00	1.00
8. Entering data	1.00	1.50

Table 3.2  
Occupation wise task importance in median values

Programming tasks	Manufac turing (n=34)	Financial (n=40)	R & D (n=43)	Admin istratio n (n=21)	Research and educational institutes (n=30)	Computer training institutes (n=33)
1. Comprehending users' problem	5.00	5.00	5.00	5.00	5.00	5.00
2. Converting specifications into detailed instruction	4.00	4.00	4.00	4.00	3.00	4.00
3. Drawing Program Flowchart	3.00	3.50	3.00	2.00	3.00	4.00
4. Coding into Computer languages	4.50	5.00	4.00	3.00	4.00	4.00
5. Preparing instruction for operator	3.00	4.00	3.00	3.00	2.50	2.00
6. Testing and correcting program	5.00	4.50	4.00	5.00	4.50	4.00
7. Modifying program	4.00	4.00	4.00	4.00	3.00	4.00
8. Entering data	1.00	1.00	1.00	1.00	2.00	3.00
9. Documenting program	4.00	3.00	3.00	3.00	3.00	3.00
10. Supervising juniors	3.50	3.00	3.00	3.00	3.00	2.00
11. Assisting other programmers	3.00	2.00	3.00	3.00	3.00	2.00
12. Designing computer configuration for users	1.00	1.00	2.00	2.00	3.00	2.00
13. Writing simple program to retrieve data	3.00	2.00	3.00	3.00	3.00	3.00
14. Collaborating with computer manufacturer	0.00	0.00	1.00	1.00	1.00	1.00



Table 3.3

Euclidean distance matrix

	1	2	3	4	5	6
1. Manufacturing (n=34)	0.00					
2. Financial (n=40)	2.24	0.00				
3. R & D (n=43)	2.12	2.55	0.00			
4. Administration (n=21)	2.55	3.39	1.73	0.00		
5. Research and educational institutes (n=30)	3.16	3.67	2.12	2.55	0.00	
6. Computer training institutes (n=33)	3.67	3.67	2.83	3.61	2.74	0.00

Figure 3.1  
Dendrogram based on occupation wise median values  
across computer programming tasks

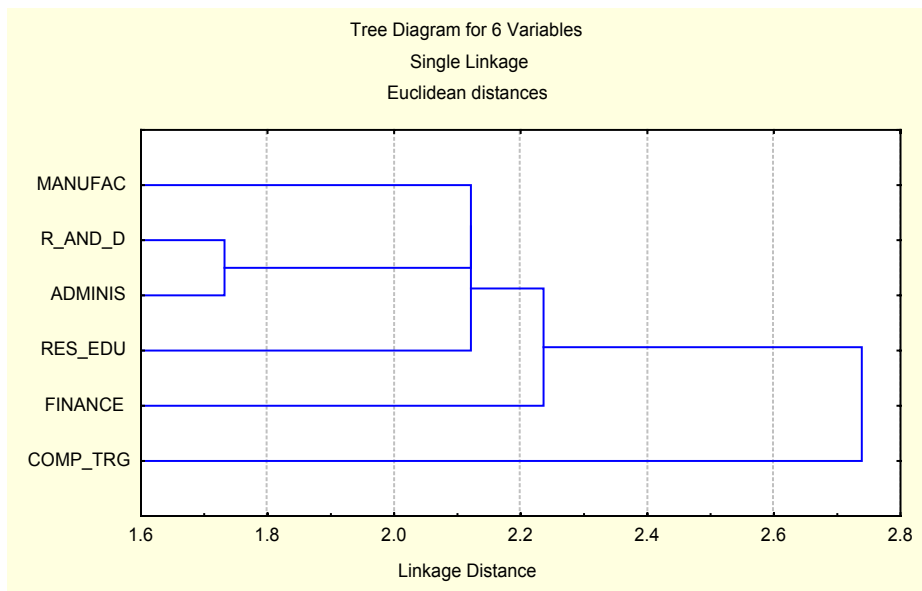


Table 3.4  
Distribution of software use (in frequency)

	Language	Object- oriente d	Package/tools	DBM	Operating System
Manufacturing(n=34)	32	6	7	16	0
Financial (n=40)	40	0	8	7	5
R & D (n=43)	41	18	21	16	3
Administration (n=21)	21	1	2	5	2
Research and educational institutes (n=30)	30	12	6	3	6
Computer training institutes (n=33)	23	16	7	9	1
Total	197	53	51	56	17

Chi square (20) = 55.36,  $p < 0.001$

Table 3.5  
Distribution of working conditions (in frequency)

	Cool	Warm	Noise
	F	f	F
Manufacturing(n=34)	7		3
Financial (n=40)	16	1	1
R & D (n=43)	11	1	4
Administration (n=21)	5		
Research and educational institutes (n=30)	14	2	
Computer training institutes (n=33)	10	2	
Total	63	6	8

Chi square (10) = 14.71, NS

Table 3.6  
Team assignment pattern across occupations (in frequency )

	Alone assignment	Team work
Manufacturing(n=34)	7	18
Financial (n=40)	8	16
R & D (n=43)	26	7
Administration (n=21)	7	10
Research and educational institutes (n=30)	8	18
Computer training institutes (n=33)	3	20

Chi square (5) = 30.56,  $p < 0.0001$

Table 3.7  
Distribution of interaction with other programmers (in frequency )

Occupations	Total	Very often	Often	Sometimes	Rarely
Manufacturing(n=34)	26	10	5	8	3
Financial (n=40)	33	6	8	14	5
R & D (n=43)	42	10	20	11	1
Administration (n=21)	18	2	4	7	5
Research and educational institutes (n=30)	26	3	11	10	2
Computer training institutes (n=33)	27	12	9	5	1
Total	172	43	57	55	17

(Chi square (15) =30.26,  $p < 0.01$ )

Table 3.8

Distribution of Occupational disorders ( in frequency )

	Total	Arth ritis	Spondilosis	Sleep disturbance	Loss of appetite	others
Manufacturing(n=34)	16	4	8	4	1	2
Financial (n=40)	17	2	12	9	1	1
R & D (n=43)	11	1	4	4	3	4
Administration (n=21)	7		3	2	1	3
Research and educational institutes (n=30)	9	2	4	3	4	1
Computer training institutes (n=33)	17		7	6	6	
Total	77	9	38	28	16	11

Chi square (20) = 28.32, NS

## RESULTS

### **Computer programming task taxonomy**

#### **A : Theoretical consideration**

Task taxonomy refers to classification or grouping of tasks into a meaningful structure. It is not mere a list of labels with semantic definition: it also has inner syntactic structure. Fleishman (1982) suggested four approaches to task classification : behavior description (detailing specific unit of behavior), behavior requirements (the internal process that are required to accomplish a task), ability requirements (physical and cognitive abilities that must be present for successful completion), and task characteristics (classify those parts of a task that have a casual relationship to performance). Factor analysis is the suitable statistical tool for taxonomic researches (Hemphill , 1960; Tornow and Pinto , 1976). Hemphill (1960) classified the management job using factor analysis. Tornow and Pinto (1976) developed a model of managerial job taxonomy through factor analytical study. Factor analysis is meant for reducing the number of variables in a set or group of measures by taking into account the overlap or correlation among the measures (Chadda, 1996; Kothari,1998). It was assumed that the results would reveal classification of programming tasks which are least overlapping and meaningful. Secondly, eigen values, the output of factor analysis could help in understanding weightage on specific programming task for job evaluation and selection of the programming personnel

#### **B: Results**

Table 4.1 shows significant correlation among 14 computer programming tasks suggesting existence of common underlying dimensions.

*Factor analysis*

Using Principal component method with varimax rotation four factors were extracted (Table 4.2) .

*Factor-1 (Program writing and testing)*

Factor-1 accounted for 23.75% variance. It has positive loading on tasks like coding into computer language (factor loading = 0.59), preparation of instruction for operator(factor loading=0.49),testing and correcting program (factor loading=0.82) , modifying program(factor loading =0.70)and documenting program(factor loading=0.41)

*Factor-2 (Human Relations)*

Factor-2 accounted for 15.46% variance. It has positive loading on the tasks namely supervising juniors(factor loading=0.82), assisting other programmers (factor loading 0.83), designing computer configuration for users (factor loading=0.60) and collaborating with computer manufacturers (factor loading =0.51).

*Factor 3-(Data analysis)*

Factor-3 accounted for 9.12% variance. It has positive loading on the tasks namely drawing program flowchart(factor loading=0.49), data entry (factor loading=0.80), and writing simple program to retrieve data (factor loading = 0.51).

*Factor-4 (User satisfaction):*

Factor-4 accounted for 8.83% variance. It has positive loading on the tasks such as comprehending users' problem (factor loading = 0.79) and converting specification into detailed instructions (factor loading = 0.77)

To sum up, results support basic assumption that the computer programming tasks are multifactorial in nature. Four computer programming task factors were extracted from this study. They were program writing and testing, Human relations, Data analysis and user satisfaction.

#### *Factor scores*

The factor loadings higher than 0.40 were considered for computing factor scores of each case according to the following formula:

$$\text{Factor score I} = .59 * I_4 + .49 * I_5 + .82 * I_6 + .7 * I_7 + .41 * I_9$$

$$\text{Factor score II} = .82 * I_{10} + .83 * I_{11} + .6 * I_{12} + .51 * I_{14}$$

$$\text{Factor score III} = .49 * I_3 + 0.8 * I_8 + .51 * I_{13}$$

$$\text{Factor score IV} = .79 * I_1 + .77 * I_2$$

#### *Differences across occupations*

Table 4.3 shows no significant differences in importance on three computer programming task factors – program writing and testing ( $F(5,195) = 1.9$ , NS), human relations ( $F(5,195) = 1.58$ , NS) and user satisfaction ( $F(5,195) = 0.323$ , NS). Importance on data analysis differs significantly ( $F(5,195) = 2.33$ ,  $p < 0.04$ ) across occupations. Programmers in computer training institutes gave more importance on data analysis than other occupations.

Table 4.1 :  
Correlation matrix of programming tasks (N=201)

Items	Mdn	Q	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	5	1	1.00													
2	4	2	<b>0.46</b>	1.00												
3	3	2	0.13	<b>0.16</b>	1.00											
4	4	2	0.29	0.13	<b>0.32</b>	1.00										
5	3	2	<b>0.17</b>	<b>0.15</b>	<b>0.23</b>	<b>0.23</b>	1.00									
6	5	1	<b>0.16</b>	<b>0.14</b>	<b>0.19</b>	<b>0.39</b>	<b>0.30</b>	1.00								
7	4	2	<b>0.19</b>	<b>0.21</b>	<b>0.22</b>	<b>0.35</b>	<b>0.26</b>	<b>0.53</b>	1.00							
8	1	3	-0.06	0.10	<b>0.21</b>	0.11	<b>0.14</b>	0.04	0.07	1.00						
9	3	2	<b>0.26</b>	0.10	<b>0.31</b>	<b>0.22</b>	<b>0.31</b>	<b>0.23</b>	<b>0.19</b>	<b>0.26</b>	1.00					
10	3	2	0.08	<b>0.15</b>	0.02	<b>-0.14</b>	<b>0.14</b>	-0.02	-0.01	0.05	<b>0.15</b>	1.00				
11	3	2	0.13	0.06	0.09	-0.02	<b>0.17</b>	0.12	<b>0.15</b>	-0.04	<b>0.24</b>	<b>0.56</b>	1.00			
12	2	3	<b>0.21</b>	<b>0.14</b>	0.13	0.05	<b>0.16</b>	-0.08	0.02	0.13	<b>0.18</b>	<b>0.37</b>	<b>0.38</b>	1.00		
13	3	2	<b>0.28</b>	<b>0.26</b>	<b>0.18</b>	<b>0.26</b>	<b>0.31</b>	<b>0.18</b>	<b>0.36</b>	<b>0.28</b>	<b>0.21</b>	0.03	0.13	<b>0.33</b>	1.00	
14	1	2	0.11	<b>0.14</b>	0.07	-0.10	0.09	<b>-0.17</b>	0.00	<b>0.14</b>	0.10	<b>0.30</b>	<b>0.26</b>	<b>0.51</b>	<b>0.21</b>	1.00

N.B: Correlation coefficients greater or equal to 0.14 are significant at .05 level and those greater or equal to 0.17 are significant at 0.01 level



Table 4.2 :  
Factor analysis of programming tasks

	Program writing and testing	Human Relations	Data analysis	User satisfaction
1. Comprehending users' problem	0.22	0.09	-0.04	<b>0.79</b>
2. Converting specifications into detailed instruction	0.10	0.04	0.04	<b>0.77</b>
3. Drawing Program Flowchart	0.37	0.02	<b>0.49</b>	0.05
4. Coding into Computer languages	<b>0.59</b>	-0.21	0.23	0.26
5. Preparing instruction for operator	<b>0.49</b>	0.23	0.31	0.07
6. Testing and correcting program	<b>0.82</b>	-0.02	-0.05	0.04
7. Modifying program	<b>0.70</b>	0.03	0.07	0.21
8. Entering data	-0.01	-0.06	<b>0.80</b>	-0.08
9. Documenting program	<b>0.41</b>	0.26	0.43	0.01
10. Supervising juniors	0.00	<b>0.82</b>	-0.04	0.01
11. Assisting other programmers	0.23	<b>0.83</b>	-0.07	-0.02
12. Designing computer configuration for users	-0.15	<b>0.60</b>	0.37	0.31
13. Writing simple program to retrieve data	0.22	0.08	<b>0.51</b>	0.44
14. Collaborating with computer manufacturer	-0.33	<b>0.51</b>	0.36	0.28
Eigen value	3.324921	2.1634186	1.2770922	1.2359846

Table 4.3

One way ANOVA in programming task factors across different occupations

		Program writing and testing	Human relations	Data analysis	User satisfaction
Manufacturing (n=34)	Mean	11.29	6.4	4.24	5.9
	SD	2.54	3.15	2.04	1.92
Financial (n=40)	Mean	11.64	5.3	3.98	6.19
	SD	2.51	3.35	1.69	1.63
R & D (n=43)	Mean	11.15	6.8	4.21	6.23
	SD	2.04	3.32	2.04	1.6
Administration (n=21)	Mean	10.3	6.83	3.46	6.13
	SD	2.88	3.37	1.93	1.48
Research and educational institutes (n=30)	Mean	10.3	7.22	4.22	6.14
	SD	2.95	3.22	1.95	1.61
Computer training institutes (n=33)	Mean	10.18	6.08	5.19	5.82
	SD	3.01	3.28	2.12	2.2
Total	Mean	10.89	6.39	4.25	6.08
	SD	2.65	3.3	1.99	1.75
	F	1.9	1.58	2.33	0.323
	(5,195)				
	Sig. Level	Ns	Ns	p<0.04	Ns

:

## RESULTS

### **Relative importance of Aptitudes**

#### **A: Theoretical considerations**

With rapid and diversified application of computer, vocational counselors tended to think which aptitudes were relatively more important for computer programmers of which occupations. Prior studies on aptitude assessment were more concerned with aptitudes of system engineers (Toshima,1993), teachers (Kurkreti,1993 and 1985,Bisht,1982; Raina,1981), science students(Sharma and Shukla, 1992; Mehta, 1982; Chatterjee and Mukerjee,1978; Agarwal,1977; Dev, 1968), manufacturing technical personnel (Mohan and Brar, 1988), office workers (Naik,1973), engineers (Deb,1973), special class railway apprentices (Avtar,1965) and paper folders(Snaw,1970).Comparatively least attention (Science research associates, 1967; Dutta Roy and Mallik, 2000) was paid to determine aptitudes of computer programmers.

Programmers are working in different occupational environments - financial institutes, research and development organizations, training institutes, manufacturing organizations. In different set ups, programmers are asked to do different functions. For example, programmers in bank concentrate on development of software for transaction of money and maintenance of inventory whereas programmers in educational institutes concentrate on teaching a set of selected programs. Therefore, there might be differences in requirements of aptitudes from the computer programmers in different occupations. Since, importance of aptitudes varies across different occupations of computer

programming, it is important to identify relatively more important aptitudes across the occupations. Determining relative importance of individual characteristics helps in saving more time for collection of data related to selection and vocational guidance (Dutta Roy, 1994).

## **B: Results**

### *Reliability of Judgements*

Reliability among responses of the judges were estimated using quartile deviations of responses for each item of checklist. It was noted that quartile deviations were within the range of 0.5 to 1.00 and it never crossed unity. The deviations for only 14 items were above 0.50. This suggests very high reliability in judgements of the judges.

### *Relative importance of aptitudes*

Disregarding occupational differences, five specific aptitudes namely comprehension, inference, expressional fluency, associational fluency and alternative usage ( Table 5.1 ) received more importance. Median values for all the above aptitudes were 4 with SD 0.50. These five aptitudes are the constituents of abstract reasoning. Findings are consistent with prior study conducted by Dutta Roy and Mallik (2000). They noted that programmers gave relatively more importance on abstract reasoning for success in computer programming. In their study, they failed to specify which specific constituents of abstract reasoning were more important. This study focused on those specific aptitudes.

*Rank order correlations*

Rank order correlations (Table 5.2) were made to determine extent of relationship in relative importance of aptitudes across occupations. The high median rank order correlation coefficient ( $\rho(40) = 0.82, p < 0.0001$ ) suggests very high relationship in relative importance of aptitude profiles among six different occupations. Correlation matrix shows that aptitude profile of administration was more close ( $\rho(40) = 0.90, p < 0.0001$ ) to that of R & D and aptitude profile of manufacturing organization was more close ( $\rho(40) = 0.69, p < 0.0001$ ) to that of manufacturing organizations. Such relative closeness suggests possible clusters among the six occupations in terms of their relative importance of aptitude profiles.

*Cluster analysis*

Single linkage method of cluster analysis was made using median values aptitudes of six occupations as inputs for analysis. Initial Euclidean distance matrix (Table 5.3) supports results of rank order correlations. Three broad clusters were noted. Cluster 1 included two occupations - manufacturing and computer training institutes. Cluster 2 included three occupations namely, administration, research and educational institutes and research and development organizations. Financial institute was widely apart from the two clusters and it belonged to cluster 3. Median test shows significant difference between cluster 1 and 2 (Chisquare (3) = 25.18,  $p < 0.0001$ ), between cluster 1 and 3 (Chisquare (3) = 9.59,  $p < 0.02$ ) but was not between 2 and 3 (Chisquare (3) = 7.35, NS).

*Cluster wise aptitude importance :*

Cluster 1 gave more importance on comprehension, arithmetic calculations, inference, expressional fluency, associational fluency and alternative usage. On the other hand cluster 2 gave more priorities on 4 specific aptitudes as comprehension, inference, expressional fluency and alternate usage. Cluster 3 gave more priorities on comprehension, absurdities, syllogistic reasoning , inference and associational fluency (Table 5.4)

Table 5.1

Relative importance of Aptitudes across Occupations

Id	Total		Manufacturing		Financial		R&D		Administration		R and E		CTI	
	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD
Comprehension	4.00	0.50	4.00	0.50	4.00	0.00	4.00	0.00	4.00	0.00	4.00	0.00	4.00	0.50
Absurdities	3.00	0.50	3.00	0.88	4.00	0.50	3.00	0.50	3.00	1.00	4.00	1.00	3.00	0.50
Verbal Relationns	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50
Sylogistic Reasoning	3.00	0.50	3.00	0.50	4.00	0.50	3.00	0.75	3.00	0.00	3.00	0.50	3.00	1.00
Typing speed	2.00	0.50	2.00	0.50	2.00	0.50	2.00	0.50	2.00	0.00	2.00	0.25	2.00	0.50
Colour recognition	2.00	0.50	2.00	0.50	1.00	0.50	2.00	0.50	2.00	0.50	2.00	1.00	2.00	1.00
Verbal reasoning	3.00	1.00	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.00	3.00	0.50	3.00	0.50
Word classification	3.00	0.50	3.00	0.50	3.00	0.50	3.00	1.00	3.00	0.50	3.00	0.50	3.00	0.50
Word opposite	2.00	0.50	2.00	0.50	1.00	0.50	2.00	0.50	2.00	0.50	2.00	1.00	2.00	0.50
Vocabulary	3.00	0.50	3.00	0.50	2.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	1.00
Anagram	2.00	1.00	2.00	1.00	2.00	1.00	2.00	1.00	2.00	1.00	2.00	1.00	2.00	0.50
Number series	3.00	0.50	3.00	1.00	3.00	0.63	2.00	0.50	2.50	0.50	3.00	0.50	3.00	1.00
Equation building	3.00	1.00	3.00	1.00	3.00	1.00	3.00	0.75	3.00	1.00	3.00	1.00	3.00	0.50
Arithmetic calculations	3.00	0.50	4.00	0.50	3.00	0.50	3.00	0.50	3.00	1.00	3.00	1.00	4.00	1.00
Inference	4.00	0.50	3.00	0.50	4.00	0.50	4.00	0.50	4.00	0.50	4.00	0.50	4.00	0.50
Pattern analysis	3.00	1.00	3.00	1.00	3.00	1.00	3.00	1.00	3.00	1.00	3.00	1.00	3.00	1.00
Spatial visualisation	2.00	1.00	2.00	0.88	1.00	0.50	2.00	1.00	2.00	1.00	2.00	1.00	2.00	0.50
Causal relations	3.00	1.00	3.00	1.00	3.00	0.63	3.00	0.50	3.00	1.00	3.00	0.63	3.00	1.00
Manipulatory visualisation	2.00	1.00	2.00	1.00	1.00	0.50	2.00	1.00	2.00	1.00	2.00	1.00	2.00	0.50
Finger dexterity	2.00	1.00	2.00	1.00	2.00	0.50	2.00	0.50	2.00	0.50	2.00	1.00	2.00	1.00
Expressional fluency	4.00	0.50	3.00	0.50	3.00	0.50	4.00	0.50	3.00	0.50	3.00	0.50	4.00	0.50
Ideational fluency	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50
Associational fluency	4.00	0.5	4.00	0.50	4.00	0.50	4.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50
Alternate usage	4.00	0.50	3.00	0.50	3.00	0.50	4.00	0.50	3.50	0.50	3.00	0.50	4.00	0.50
Sketches	3.00	1.00	3.00	0.50	3.00	0.63	3.00	0.50	2.50	0.50	3.00	0.50	3.00	0.50
Matching problems	3.00	0.50	3.00	0.88	3.00	0.50	3.00	0.50	3.00	0.00	3.00	0.50	3.00	1.00
Decoration	3.00	0.50	2.00	0.50	3.00	1.00	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50

Table 5.1 (Continued)

	Total		Manufacturing		Financial		R&D		Administration		Rand E		CTI	
	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD	Mdn	QD
Quick grasping of visual details	3.00	1.00	3.00	0.50	3.00	1.00	3.00	1.00	3.00	0.00	3.00	0.50	3.00	1.00
Quick grasping of similarities	3.00	0.50	3.00	0.88	3.00	0.50	3.00	0.50	3.00	1.00	3.00	1.00	3.00	0.50
Word fluency	3.00	0.50	3.00	0.88	3.00	0.50	2.00	0.50	3.00	0.50	2.00	0.50	3.00	1.00
Coding speed	3.00	0.50	3.00	0.88	3.00	0.50	2.50	0.75	2.00	0.50	2.00	0.50	3.00	1.00
Memory for digits	2.00	0.50	3.00	0.50	2.00	0.50	2.00	0.50	2.00	0.50	2.00	0.50	3.00	0.50
Matching geometric forms	2.00	1.00	2.00	1.00	1.00	1.00	2.00	0.75	2.00	1.00	2.00	1.00	2.00	1.00
Copying	3.00	1.00	3.00	0.50	3.00	1.00	3.00	1.00	3.00	0.50	3.00	0.50	3.00	0.50
Matrices	3.00	0.50	3.00	0.50	3.00	0.63	3.00	0.50	3.00	1.00	3.00	0.63	2.00	0.50
Memory for commands	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	0.50	3.00	1.00
Memory for spatial position	2.00	0.50	3.00	0.50	2.00	0.50	2.00	0.50	2.00	0.50	3.00	0.50	2.50	0.50
Perception of fixed spatial/geometrical relation	2.00	1.00	2.00	1.00	2.00	0.50	2.00	0.50	2.00	1.00	2.00	1.00	2.00	0.50
Musical ability	1.00	0.50	1.00	0.50	1.00	0.00	1.00	0.50	2.00	0.50	2.00	0.50	1.00	0.50



Table 5.2

Spearman Rank-order correlation coefficients among different occupational groups in their judgement on relative importance of computer programming aptitudes

	Administration (n=34)	Computer training institute (n=40)	Financial (n=43)	Manufacturing (n=21)	R&D (n=30)	Research and educational institutes (n=33)
Administration	1.00	.				
Computer training institute	0.82	1.00				
Financial	0.83	0.78	1.00			
Manufacturing	0.72	0.82	0.79	1.00		
R&D	0.90	.0.82	0.81	0.73	1.00	
Research and educational institutes	0.81	0.70	0.79	0.69	0.81	1.00

\*All correlation coefficients were significant at .0001 level.

Table 5.3

Euclidean distances among different occupational groups in their judgement on relative importance of computer programming aptitudes

	1	2	3	4	5	6
1. Manufacturing	.00					
2. Finance	3.61	.00				
3. Research and Development	3.04	3.50	.00			
4. Administration	2.96	3.43	2.24	.00		
5. Research and educational institute	3.16	3.61	2.87	2.18	.00	
6. Computer training institute	2.50	3.77	2.55	2.65	3.20	.00

Figure 5.1  
Dendrogram of occupations based upon relative importance of aptitudes

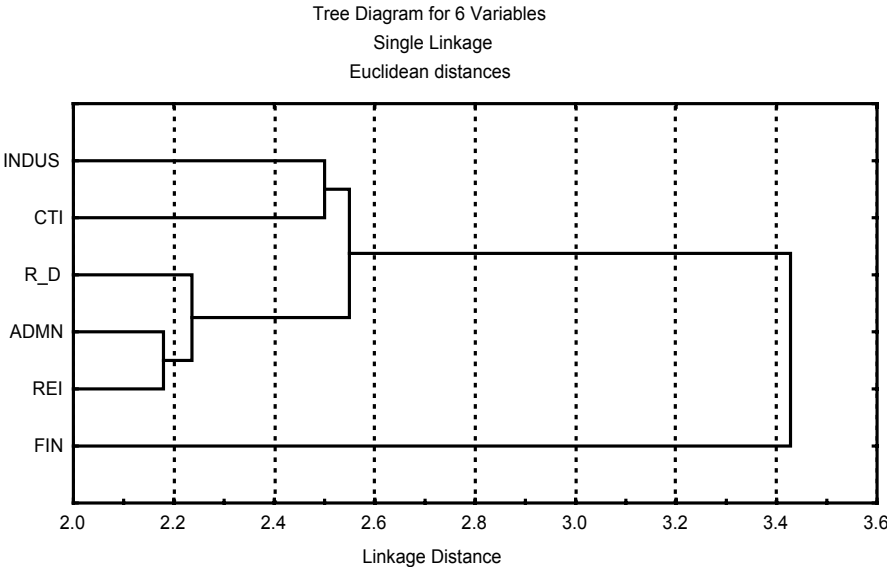


Table 5.4  
Aptitude importance across three clusters in median values

Sl. No. of items	Aptitudes	Cluster 1	Cluster 2	Cluster 3
1	Comprehension	4	4	4
2	Absurdities	3	3	4
3	Verbal Relations	3	3	3
4	Syllogistic Reasoning	3	3	4
5	Typing speed	2	2	2
6	Colour recognition	2	2	1
7	Verbal reasoning	3	3	3
8	Word classification	3	3	3
9	Word opposite	2	2	1
10	Vocabulary	3	3	2
11	Anagram	2	2	2
12	Number series	3	3	3
13	Equation building	3	3	3
14	Arithmetic calculations	4	3	3
15	Inference	4	4	4
16	Pattern analysis	3	3	3
17	Spatial visualisation	2	2	1
18	Causal relations	3	3	3
19	Manipulatory visualisation	2	2	1
20	Finger dexterity	2	2	2
21	Expressional fluency	4	4	3
22	Ideational fluency	3	3	3
23	Associational fluency	4	3	4
24	Alternate usage	4	4	3
25	Sketches	3	3	3
26	Matching problems	3	3	3
27	Decoration	3	3	3
28	Matching identical drawings	2	2	2
29	Quick grasping of visual details	3	3	3
30	Quick grasping of similarities	3	3	3
31	Word fluency	3	2	3
32	Coding speed	3	2	3
33	Memory for digits	3	2	2
34	Matching geometric forms	2	2	1
35	Copying	3	3	3
36	Matrices	3	3	3
37	Memory for commands	3	3	3
38	Memory for spatial position	3	2	2
39	Perception of fixed spatial/geometrical relation	2	2	2
40	Musical ability	1	1	1

## RESULTS

### **Relationship between task and aptitude importance**

#### **A : Theoretical considerations**

Importance on aptitude varies with its importance on different tasks. In earlier chapters, it is evident that programming tasks are multidimensional in nature and they are almost equally important for all the occupations. Secondly, aptitude importance varied across occupations. But which aptitudes are important for which programming tasks are yet unknown. Here is an attempt to understand correlations between importance on computer programming task factors and the importance on listed aptitudes.

#### **B. Results**

Tables 6.1 , 6.2, 6.3 and 6.4 represent Pearson product moment correlation coefficients between each computer programming task factor and each of the listed aptitudes across the occupations. It was noted that significant correlation coefficients between task factors and listed aptitudes did not vary significantly (Chi-square (15) = 15.47,  $p < 0.14$ ) across occupations. Disregarding occupational differences, it was noted that human relations (57.5%) and data analysis (52.5%) task factors were significantly (chi-square (3) = 15.71,  $p < 0.0013$ ) more related to the listed aptitudes than program writing and testing (32.5%) and user satisfaction (22.5%).

*Program writing and testing:*

Table 6.1 shows that program writing and testing factor was significantly related to 13 aptitudes. Of 13 aptitudes it was more related to memory for commands(0.33), copying (0.27) , syllogistic reasoning ( 0.21), typing speed (0.21), verbal relations (0.21) than quick grasping of similarities(0.20), matrices (0.20), matching problems (0.18) , coding speed (0.17), finger dexterity (0.17), comprehension (0.14), memory for spatial position (0.14) and memory for digits (0.14).

*Human Relations:*

Table 6.2 shows that the human relations task factor was significantly related to 23 aptitudes. Of 23 aptitudes it was more related to matching identical drawings (0.36), spatial visualization (0.33), musical ability (0.29), manipulatory visualization (0.28), decoration (0.28) than word opposites (0.27), pattern analysis (0.27), perception of fixed spatial and geometrical relations (0.24), quick grasping of visual details (0.24), matching geometrical forms (0.23) , quick grasping of visual details (0.24), finger dexterity (0.22), memory for spatial position (0.21), matrices (0.20), color recognition (0.20), alternate uses (0.19), sketches (0.19), arithmetic calculations (0.19), matching problems (0.18), word fluency (0.19), anagram (0.18), causal relations (0.16) and word classification (0.15).

*Data analysis*

Table 6.3 shows that the data analysis task factor was significantly related to 21 aptitudes. Of these 21 aptitudes it was more related to coding speed(0.30), finger dexterity(0.25), number series(0.25) anagram(0.24), pattern analysis(0.21), coding speed(0.21), sketches(0.20), memory for spatial position(0.20) than matching geometric

forms(0.19), color recognition (0.19), manipulatory visualization(0.19), word fluency(0.18), matching identical drawings(0.18) arithmetic calculation(0.18), perception of fixed spatial or geometrical relations(O.18) , matching problems(0.16), decoration(0.16), word classification(0.14), vocabulary(0.14), and musical ability(0.14).

*User satisfaction:*

Table 6.4 shows that user satisfaction task factor was significantly related to 9 aptitudes. Of these 9 aptitudes it was more related to spatial visualization (0.23), inference (0.22), comprehension(0.17), finger dexterity(0.16), word fluency (0.16) than coding speed(0.15), copying(0.15),color recognition(0.15) and verbal reasoning (0.14).

Analysis of Computer programming tasks and Aptitudes

Table 6.1  
Correlation coefficients between program writing and testing and aptitudes across occupations

	Manufaturing (n=34)	Financia l (n=40)	R & D (n=43)	Administra tion (n=21)	Research and Edu. (n=30)	Compu ter training (n=33)	Total (n=201)
Comprehension	0.12	0.13	0.19	0.04	<b>0.56</b>	0.02	<b>0.14</b>
Absurdities	0.22	0.28	-0.23	-0.08	0.16	-0.23	0.04
Verbal Relations	0.22	<b>0.41</b>	-0.19	0.28	<b>0.45</b>	0.15	<b>0.21</b>
Syllogistic Reasoning	0.05	<b>0.47</b>	0.02	0.04	0.20	0.23	<b>0.21</b>
Typing speed	0.13	0.08	0.27	0.09	0.29	-0.23	0.08
Colour recognition	0.32	-0.04	-0.01	0.31	0.03	0.03	0.05
Verbal reasoning	-0.24	<b>0.37</b>	-0.03	0.29	0.00	-0.24	0.01
Word classification	-0.17	0.19	0.04	0.24	0.01	-0.16	0.01
Word opposite	-0.15	0.11	0.15	0.12	0.00	0.05	0.01
Vocabulary	0.11	0.14	0.19	0.06	-0.06	-0.34	-0.03
Anagram	-0.06	0.19	0.01	-0.02	0.07	-0.03	0.01
Number series	-0.10	0.30	-0.15	0.08	-0.12	0.19	0.03
Equation building	-0.15	0.28	-0.12	-0.08	-0.19	-0.07	-0.04
Arithmetic calculations	-0.15	<b>0.34</b>	0.01	0.31	0.00	-0.17	0.06
Inference	0.16	0.28	0.01	0.06	0.28	-0.27	0.08
Pattern analysis	-0.12	0.17	0.03	0.17	0.05	0.2	0.06
Spatial visualisation	0.28	-0.30	-0.01	0.24	<b>-0.44</b>	0.07	-0.08
Causal relations	0.10	0.30	0.16	-0.03	-0.17	0.04	0.08
Manipulatory visualisation	0.22	-0.15	0.03	-0.08	-0.21	0.19	-0.03
Finger dexterity	0.26	0.14	-0.03	0.40	0.01	0.31	<b>0.17</b>
Expressional fluency	-0.02	-0.03	0.12	0.40	0.06	0.02	0.07
Ideational fluency	0.09	-0.06	-0.03	0.14	-0.24	0.16	-0.01
Associational fluency	0.00	<b>0.32</b>	0.04	<b>0.45</b>	-0.11	-0.01	0.13
Alternate usage	-0.28	0.24	-0.19	0.39	0.17	-0.09	0.00
Sketches	-0.07	0.12	0.03	0.16	-0.36	0.2	0.00
Matching problems	0.07	0.29	0.00	0.05	0.12	<b>0.36</b>	<b>0.18</b>
Decoration	0.15	0.27	0.09	0.22	0.03	0.11	<b>0.14</b>
Matching identical drawings	0.21	0.09	0.14	0.43	-0.15	0.11	0.08
Quick grasping of visual details	<b>0.40</b>	0.10	0.09	0.04	-0.08	<b>-0.37</b>	0.01
Quick grasping of similarities	0.03	<b>0.40</b>	0.23	0.10	0.37	0.03	<b>0.20</b>
Word fluency	-0.13	0.02	0.19	-0.06	0.21	0.06	0.06
Coding speed	0.15	0.01	0.18	0.09	<b>0.56</b>	0.06	<b>0.17</b>
Memory for digits	0.10	0.22	<b>0.32</b>	0.01	0.15	-0.02	<b>0.14</b>
Matching geometric forms	0.01	0.19	0.15	0.26	-0.14	-0.09	0.03
Copying	-0.10	<b>0.50</b>	-0.08	0.32	<b>0.51</b>	0.27	<b>0.27</b>
Matrices	0.11	<b>0.40</b>	0.10	0.06	0.03	<b>0.37</b>	<b>0.20</b>
Memory for commands	0.13	<b>0.44</b>	<b>0.39</b>	0.18	<b>0.56</b>	0.17	<b>0.33</b>
Memory for spatial position	-0.04	0.14	0.16	0.40	0.10	0.21	<b>0.14</b>
Perception of fixed spatial/geometrical relation	-0.01	0.02	0.11	0.41	-0.18	0.09	0.03
Musical ability	-0.19	-0.18	0.29	0.08	-0.08	-0.08	-0.06

Analysis of Computer programming tasks and Aptitudes

Table 6.2  
Correlation coefficients between importance on  
Human relations and aptitudes across occupations

	Manufac turing (n=34)	Financi al (n=40)	R & D (n=43)	Admini stration (n=21)	Research and Edu (n=30)	Computer training (n=33)	Total (n=201)
Comprehension	0.15	-0.08	0.11	0.32	-0.12	0	0.07
Absurdities	-0.16	0.02	-0.05	0.13	-0.18	0.02	-0.05
Verbal Relationns	-0.20	-0.02	0.14	0.39	0.16	0.01	0.06
Syllogistic Reasoning	-0.27	-0.10	0.27	0.11	0.23	0.28	0.05
Typing speed	-0.18	0.20	0.18	0.31	0.16	-0.24	0.05
Colour recognition	<b>0.36</b>	0.07	0.24	0.13	0.20	0.19	<b>0.20</b>
Verbal reasoning	-0.29	-0.03	0.05	0.14	<b>0.38</b>	-0.13	-0.03
Word classification	0.11	<b>0.35</b>	0.11	0.18	-0.01	0.12	<b>0.15</b>
Word opposite	-0.16	<b>0.37</b>	<b>0.35</b>	0.11	<b>0.43</b>	0.25	<b>0.27</b>
Vocabulary	0.12	0.21	0.28	0.12	-0.04	-0.15	0.11
Anagram	0.12	0.27	0.19	-0.05	<b>0.39</b>	0.06	<b>0.18</b>
Number series	0.16	0.07	0.09	0.00	0.21	0.21	0.12
Equation building	0.23	0.17	-0.06	0.27	0.17	0.04	0.13
Arithmetic calculations	0.16	0.07	0.20	0.35	0.34	0.2	<b>0.19</b>
Inference	0.07	0.25	0.09	<b>0.48</b>	-0.01	0.07	0.13
Pattern analysis	<b>0.44</b>	0.28	0.25	0.26	0.30	0.14	<b>0.27</b>
Spatial visualisation	<b>0.44</b>	<b>0.42</b>	0.30	0.25	0.16	0.27	<b>0.33</b>
Causal relations	0.07	0.20	0.12	0.20	0.22	0.32	<b>0.16</b>
Manipulatory visualisation	0.32	<b>0.44</b>	0.19	0.03	0.14	<b>0.43</b>	<b>0.28</b>
Finger dexterity	0.11	0.04	<b>0.34</b>	<b>0.53</b>	<b>0.42</b>	0.16	<b>0.22</b>
Expressional fluency	<b>-0.35</b>	0.00	-0.17	0.16	0.37	0.26	0.05
Ideational fluency	0.16	0.11	0.18	-0.22	0.24	-0.01	0.10
Associational fluency	-0.22	0.09	0.01	0.13	-0.21	0.1	-0.05
Alternate usage	-0.04	0.29	<b>0.40</b>	0.16	0.09	0.17	<b>0.19</b>
Sketches	0.34	0.22	0.20	-0.12	0.33	0.14	<b>0.19</b>
Matching problems	-0.17	0.13	<b>0.39</b>	-0.03	0.29	<b>0.48</b>	<b>0.18</b>
Decoration	-0.06	<b>0.44</b>	0.28	0.34	0.31	<b>0.46</b>	<b>0.28</b>
Matching identical drawings	<b>0.52</b>	<b>0.45</b>	<b>0.31</b>	0.02	0.27	<b>0.54</b>	<b>0.36</b>
Quick grasping of visual details	0.03	<b>0.32</b>	<b>0.45</b>	0.28	<b>0.38</b>	-0.02	<b>0.24</b>
Quick grasping of similarities	0.11	<b>0.34</b>	0.23	0.17	0.30	0.22	<b>0.22</b>
Word fluency	-0.08	0.15	<b>0.43</b>	0.39	0.19	0.19	<b>0.19</b>
Coding speed	0.01	-0.06	<b>0.39</b>	0.35	0.24	-0.16	0.10
Memory for digits	0.30	-0.05	0.03	0.29	<b>0.50</b>	-0.11	0.13
Matching geometric forms	<b>0.58</b>	0.09	0.23	0.15	0.36	0.01	<b>0.23</b>
Copying	-0.21	0.09	0.28	0.21	0.07	0.1	0.08
Matrices	0.16	0.02	0.15	<b>0.51</b>	0.22	0.31	<b>0.20</b>
Memory for commands	-0.15	0.03	-0.13	0.42	-0.07	0.15	0.04
Memory for spatial position	0.03	0.11	0.07	<b>0.56</b>	<b>0.40</b>	0.26	<b>0.21</b>
Perception of fixed spatial/geometrical relation	<b>0.51</b>	<b>0.34</b>	0.24	0.21	0.19	-0.04	<b>0.24</b>
Musical ability	0.29	<b>0.36</b>	<b>0.31</b>	0.22	0.18	0.24	<b>0.29</b>



Analysis of Computer programming tasks and Aptitudes

**Table 6.3**  
Correlation coefficients between importance on  
Data analysis and aptitudes across occupations

	Manuf acturin g (n=34)	Financi al (n=40)	R & D (n=43)	Admini stration (n=21)	Resear ch and educati onal institut es (n=30)	Compu ter trainin g institut es (n=33)	Total (n=201 )
Comprehension	0.08	0.17	0.00	0.17	<b>0.40</b>	-0.25	-0.01
Absurdities	0.13	0.22	-0.29	-0.24	-0.25	-0.31	-0.11
Verbal Relationns	0.09	<b>0.32</b>	-0.15	0.20	0.20	-0.25	0.03
Syllogistic Reasoning	0.02	0.12	-0.05	0.30	<b>0.40</b>	-0.21	0.05
Typing speed	0.03	-0.07	0.06	0.31	<b>0.51</b>	-0.06	0.09
Colour recognition	0.28	-0.05	-0.04	0.41	0.18	<b>0.35</b>	<b>0.19</b>
Verbal reasoning	-0.33	0.26	-0.16	-0.13	0.20	<b>0.44</b>	0.04
Word classification	0.12	0.29	0.12	0.12	0.22	0	<b>0.14</b>
Word opposite	<b>-0.37</b>	-0.01	<b>0.33</b>	0.39	0.14	0.33	0.13
Vocabulary	0.29	-0.12	0.02	0.33	0.15	0.28	<b>0.14</b>
Anagram	0.15	0.14	0.10	0.45	0.29	<b>0.36</b>	<b>0.24</b>
Number series	0.29	0.10	0.18	<b>0.48</b>	0.15	0.24	<b>0.25</b>
Equation building	0.08	0.19	-0.03	0.37	0.17	-0.03	<b>0.14</b>
Arithmetic calculations	-0.06	0.27	0.15	0.44	0.28	-0.03	<b>0.17</b>
Inference	<b>0.35</b>	0.16	0.20	-0.07	0.11	-0.1	0.13
Pattern analysis	0.14	<b>0.32</b>	<b>0.33</b>	0.42	0.15	-0.2	<b>0.21</b>
Spatial visualisation	0.33	-0.03	0.23	<b>0.59</b>	0.04	-0.21	0.14
Causal relations	0.10	0.07	0.03	0.14	0.10	-0.18	0.04
Manipulatory visualisation	0.33	0.15	0.28	0.35	0.18	-0.3	<b>0.19</b>
Finger dexterity	<b>0.35</b>	-0.01	0.21	0.32	0.32	0.3	<b>0.25</b>
Expressional fluency	-0.25	-0.09	0.05	-0.01	-0.10	-0.15	-0.06
Ideational fluency	0.32	0.27	-0.09	0.07	-0.01	-0.04	0.10
Associational fluency	0.26	0.06	-0.08	-0.03	-0.11	-0.04	0.01
Alternate usage	-0.01	0.14	-0.17	0.09	0.20	-0.04	0.02
Sketches	0.11	<b>0.43</b>	0.14	0.32	-0.09	0.14	<b>0.20</b>
Matching problems	<b>0.36</b>	0.25	0.07	0.20	0.19	0.02	<b>0.16</b>
Decoration	0.07	<b>0.42</b>	<b>0.31</b>	0.34	0.06	-0.16	<b>0.16</b>
Matching identical drawings	0.13	0.30	<b>0.42</b>	<b>0.48</b>	0.20	<b>-0.36</b>	<b>0.18</b>
Quick grasping of visual details	<b>0.37</b>	0.09	<b>0.33</b>	0.21	0.13	-0.23	<b>0.15</b>
Quick grasping of similarities	0.20	<b>0.37</b>	0.06	0.29	0.30	-0.33	0.12
Word fluency	-0.12	0.25	<b>0.33</b>	0.28	0.02	0.3	<b>0.18</b>
Coding speed	0.14	0.05	0.30	0.22	<b>0.56</b>	0.33	<b>0.30</b>
Memory for digits	0.19	0.12	<b>0.35</b>	0.21	0.29	0.05	<b>0.21</b>
Matching geometric forms	0.21	0.19	0.25	0.42	0.31	-0.18	<b>0.19</b>
Copying	-0.10	<b>0.32</b>	0.13	0.39	0.19	-0.2	0.10
Matrices	0.18	-0.03	0.26	0.27	0.13	-0.19	0.08
Memory for commands	0.12	0.23	0.24	0.40	0.33	-0.15	0.12
Memory for spatial position	0.14	0.30	0.26	0.34	<b>0.43</b>	-0.2	<b>0.20</b>
Perception of fixed spatial/geometrical relation	0.07	0.11	0.26	<b>0.49</b>	0.19	-0.08	<b>0.17</b>
Musical ability	-0.06	0.04	<b>0.35</b>	0.41	<b>0.42</b>	-0.11	<b>0.14</b>

Analysis of Computer programming tasks and Aptitudes

**Table 6.4**  
Correlation coefficients between importance on  
User satisfaction and aptitudes across occupations

	Manufacturing (n=34)	Financial (n=40)	R & D (n=43)	Administration (n=21)	Research & Edu(n=30)	Computer training (n=33)	Total (n=201)
Comprehension	0.25	0.17	0.10	0.43	<b>0.46</b>	-0.07	<b>0.17</b>
Absurdities	-0.03	0.06	-0.15	0.32	0.16	0	0.04
Verbal Relations	-0.09	0.14	-0.25	0.21	0.27	-0.23	-0.04
Syllogistic Reasoning	0.32	0.04	0.07	0.29	0.10	-0.01	0.11
Typing speed	-0.02	0.13	0.22	0.08	0.15	-0.08	0.06
Colour recognition	0.17	0.05	<b>0.39</b>	0.17	0.07	0.15	<b>0.15</b>
Verbal reasoning	0.11	0.13	0.25	<b>0.55</b>	-0.10	0.13	<b>0.14</b>
Word classification	-0.13	0.31	0.08	0.23	0.04	-0.02	0.07
Word opposite	0.04	-0.03	0.07	0.15	-0.11	-0.1	-0.01
Vocabulary	<b>0.39</b>	0.00	<b>0.39</b>	0.33	-0.09	-0.24	0.09
Anagram	-0.27	0.07	0.10	-0.07	-0.18	0.25	0.00
Number series	-0.10	0.04	-0.05	-0.04	-0.10	-0.01	-0.05
Equation building	-0.06	0.08	-0.20	-0.07	-0.26	0.16	-0.05
Arithmetic calculations	0.05	-0.13	-0.09	0.02	-0.16	0.26	0.00
Inference	0.29	0.24	0.21	0.43	0.10	0.15	<b>0.22</b>
Pattern analysis	0.00	0.16	0.04	0.20	-0.08	0.02	0.05
Spatial visualisation	<b>0.46</b>	0.14	<b>0.40</b>	0.18	-0.23	0.31	<b>0.23</b>
Causal relations	0.10	0.20	0.07	<b>0.56</b>	-0.28	0.15	0.10
Manipulatory visualisation	<b>0.45</b>	-0.03	-0.03	0.14	-0.16	0.15	0.09
Finger dexterity	<b>0.49</b>	-0.07	0.18	0.38	-0.06	0.11	<b>0.16</b>
Expressional fluency	-0.04	0.26	-0.28	-0.04	-0.21	-0.12	-0.06
Ideational fluency	-0.08	<b>0.48</b>	0.10	-0.14	-0.34	0.21	0.07
Associational fluency	0.12	-0.02	-0.09	0.28	-0.05	0.3	0.10
Alternate usage	0.14	0.27	0.03	0.01	0.08	0.01	0.10
Sketches	-0.17	-0.13	<b>0.41</b>	0.27	<b>-0.51</b>	0.24	0.03
Matching problems	-0.24	0.01	0.30	-0.04	-0.08	0.26	0.06
Decoration	-0.16	0.05	0.18	0.09	-0.20	0.28	0.05
Matching identical drawings	0.29	-0.20	<b>0.32</b>	0.14	-0.22	0.12	0.07
Quick grasping of visual details	<b>0.41</b>	-0.16	0.22	0.03	-0.29	-0.1	0.03
Quick grasping of similarities	0.03	0.09	0.17	-0.07	0.23	0.12	0.11
Word fluency	0.33	0.16	0.28	-0.12	0.17	0.07	<b>0.16</b>
Coding speed	<b>0.51</b>	-0.10	<b>0.32</b>	-0.06	0.29	-0.05	<b>0.15</b>
Memory for digits	0.05	-0.20	-0.09	-0.18	-0.10	0	-0.08
Matching geometric forms	-0.01	-0.17	0.11	0.05	-0.16	0.09	-0.02
Copying	-0.07	-0.12	0.15	0.31	<b>0.43</b>	<b>0.4</b>	<b>0.15</b>
Matrices	0.29	-0.22	-0.01	0.36	-0.11	0.17	0.05
Memory for commands	0.18	0.26	-0.03	0.26	<b>0.44</b>	-0.2	0.11
Memory for spatial position	-0.25	-0.01	0.18	0.36	-0.03	0.29	0.07
Perception of fixed spatial/geometrical relation	0.06	-0.04	0.16	0.16	-0.17	0.04	0.02
Musical ability	-0.17	0.13	0.17	0.10	-0.06	-0.01	0.02

Table 6.5  
No of aptitudes significantly related to programming task factors (n=201)

Program writing and testing	13	32.5
Human relations	23	57.5
Data analysis	21	52.5
User satisfaction	9	22.5

chi-square (3) = 15.71, p<0.0013

Table 6.6  
No of aptitudes significantly related to programming task factors among six occupations

	Manufacturing (n=34)	Financial (n=40)	R & D (n=43)	Administration (n=21)	Research and Edu. (n=30)	Computer training (n=33)
Program writing and testing	1	9	2	1	5	2
Human relations	6	10	9	4	7	4
Data analysis	4	6	8	4	6	3
User satisfaction	6	1	6	2	3	1

### **Description of Computer aided Aptitude Test Battery**

The test battery consists of 11 sub tests. Computer algorithm used for development of test battery is shown in the appendices. Figure 1 shows flowchart for algorithm development of 10 subtests (absurdity, anagram, vocabulary, number series, inference, arithmetical reasoning, coding and decoding, memory for commands, comprehension, and creativity ). And Figure 2 represents flow chart for short term memory test. Computer algorithm of all test battery included in appendix B.

#### *Absurdity*

This test measures ability to understand and identify things which are false or against reasons. Test consists of 15 questions. In each question, there are 4 items, 3 of these items are alike while the fourth differs from the others in some respect. Candidate is required to indicate the odd item and type the answer option on the key board. Scoring is based on the number of correct responses. Average time required to finish the test is 10 minutes. Flow chart of this test is given in Figure 6.1.

#### *Anagram*

This test measures ability to arrange the mixed up letters of a word to make it meaningful. The test consists of 15 items. Each word is followed by 4 answer options. Only one is correct. The difficulty of the test is increased by increasing progressively the length of the words presented. Average time to complete the test is 10 minutes. Flow chart of this test is given in Figure 6.1

### *Vocabulary*

It intends to measure the ability to understand the meaning of given words. The test is divided into two sections – (a) antonyms and (b) synonyms. Each section consists of 15 words. Each word is followed by a number of answer options. Subject is asked to identify the correct answer option and to type it. Difficulty in both sections is increased progressively. Words presented at the beginning of each section are more familiar than words presented at the later part of that section. Average time required for the test is 15 minutes. Flow chart of this test is given in Figure 6.1

### *Number Series*

This test measures the subject's ability to identify a number appropriate to a given series in which numbers are changing in a particular order. The test consists of 20 items. 4 answer options are provided with each item. Only one is correct. The subject is to find out the correct answer choice and type it. Average time required for the test is 15 minutes. Flow chart of this test is given in Figure 6.2

### *Inference*

This test measures the ability to predict or infer the outcome of a problem. This test consists of two sections-(a) Syllogistic Reasoning and (b) Data Sufficiency. Each section consists of 15 items. Each item in section consists of two statements (propositions) and two conclusions. Subject has to derive the conclusions from the given statements with applications of rules given. In section B, the subject is required to determine whether the data provided with reference to

a given question is sufficient to answer the question. Average time required to finish the test is 30 minutes. Flow chart of this test is given in Figure 6.1

### *Numerical Reasoning*

This test measures the ability to understand a problem and to perform basic arithmetical operations accurately and quickly. This test consists of 40 items. Each item is followed by 4 answer options. Subject is required to type the correct answer option. Average time required to finish the test is 30 minutes. Flow chart of this test is given in Figure 6.2

### *Coding and decoding*

It intends to measure the ability to recognize the commonness between 2 or more problems. There are 15 items in this test. Each item consists of combinations of digits and letters. Subject is to find out the right combination from the answer options, considering the similarity between the combination of letters and digits and type the answer option. Difficulty level of the items is increased by increasing the number of letters and digits. Average time required to complete the test is 15 minutes. Flow chart of this test is given in Figure 6.1

### *Memory for commands*

This test measures the subject's ability to memorize commands and reproduce them on demand. This test is made of 3 sections each consisting of 10 items. Section A consists of paired associated commands and their meanings. Section B consists of paired associated commands only while Section C consists of commands and their symbolic representations. In each section the subject is shown the paired associated items for a fixed time. After that he is shown only one

number of a pair and is required to type the other member. Average time required for the test is 20 minutes. Flow chart of this test is given in Figure 6.1

#### *Short term memory test*

This test measures subject's capacity to remember the length of digits after presentation for 2 seconds. The program is based on JAVA. Program stops after twice failure in remembering the correct number of digits presented. Flow chart of this test is given in Figure 6.3

#### *Comprehension*

This test measures the ability to understand the demands of a problem and the ability to use words and sentences logically. The test consists of 3 sections- (a)Comprehension from passage (b)Verbal Reasoning 1 and (c) Verbal Reasoning II. While Section B and Section C consist of items of the same type, questions in Section C is more difficult than those in section B. Section A and B consists of 5 items each. Section C consists of 15 items. Average time required for the test is 25 minutes. Flow chart of this test is given in Figure 6.1

#### *Creativity*

This test consists of 5 sections. Section A(Word Fluency) measures the ability to use a word in different situations. The time limit for this section is 10 minutes. Section B(Associational Fluency) calls for single, meaningful words to satisfy a relationship. This section consists of 4 items. The time limit for each item is 2 minutes. Section C (Expressional Fluency) requires the subject to Produce connected discourse. The first part of this section asks the subject to construct meaningful sentences with words whose initial letters are specified. The second

part of this section asks the subject to construct meaningful sentences with 2 or more given words. Section C consists of 10 items. The time limit for each item is 2 minutes. Section D (Alternate Uses) measures the flexibility in the subject's thought process. The subject is asked to list as many uncommon uses he can think of a common household item. This section consists of 5 items. The time limit for each item is 2 minutes. Section E (Originality) measures the ability to produce novel responses using the standard of cleverness. The subject is asked to suggest an original title to a well-known story. The time limit is 2 minutes. Flow chart of this test is given in Figure 6.1



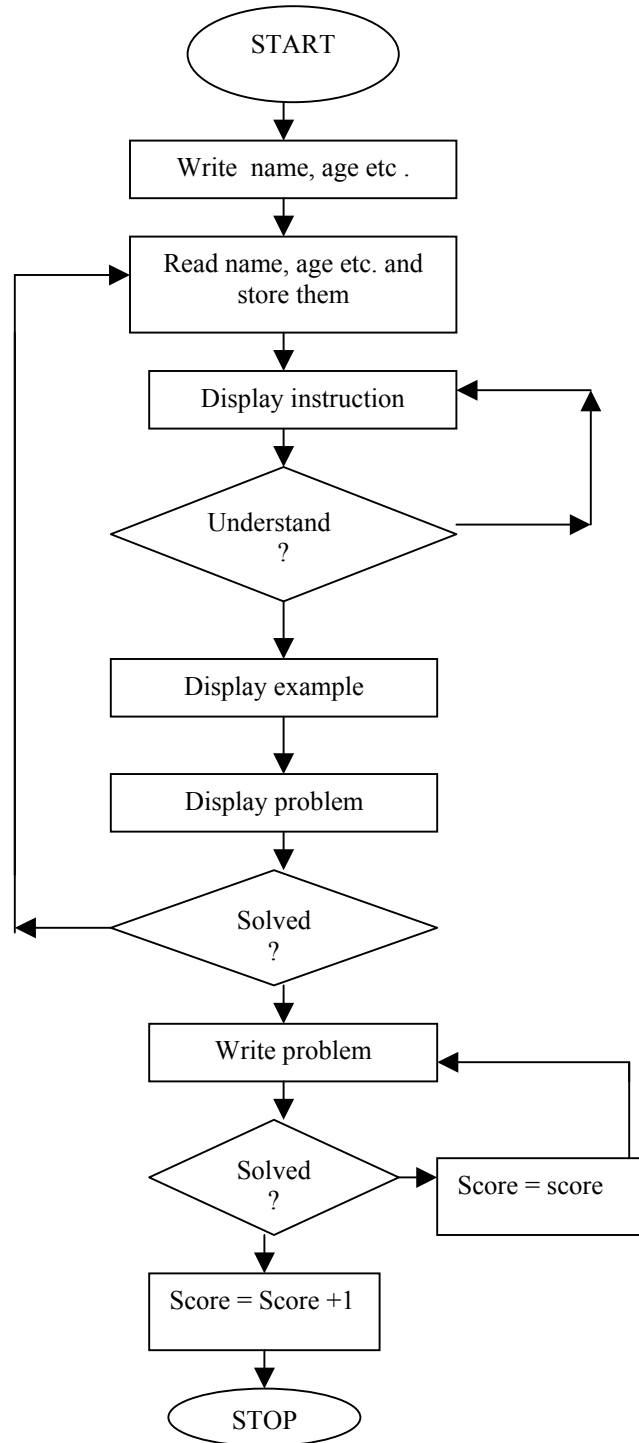


Figure 6.1  
Flow chart for absurdity, anagram, coding and decoding, inference, comprehension,  
vocabulary aptitude test battery

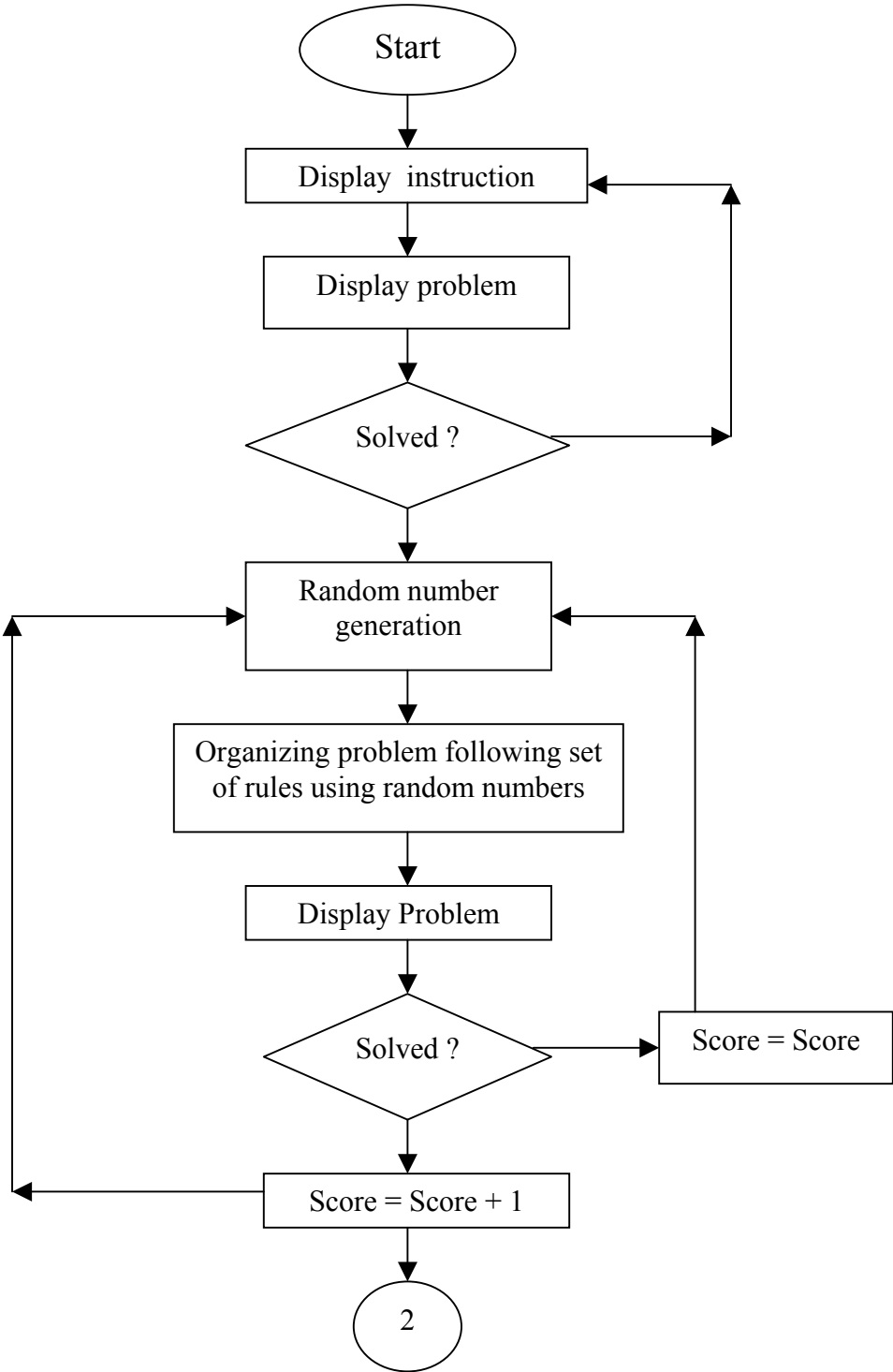


Figure 6.2  
Flow chart for number series and numerical reasoning tests

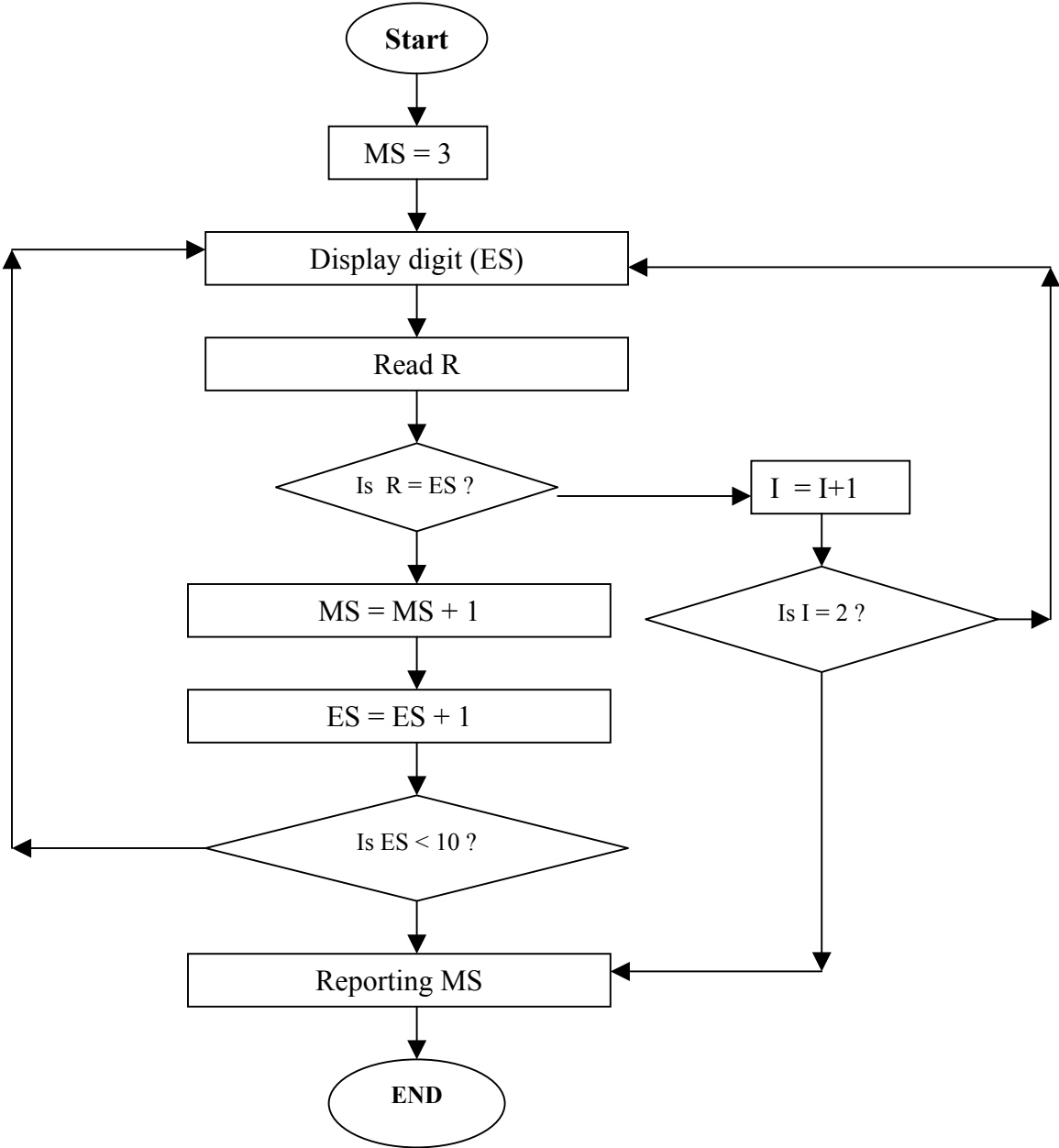


Figure 6.3  
Flow chart for short term memory tests

## SUMMARY FINDINGS

1. The five programming tasks were relatively more important to the computer programmers. They were (a) comprehending users' problems , (b) testing and correcting programs, (c) converting specifications into detailed instructions, (d) coding into computer languages and (e) modifying program .
2. Cluster analysis classified median values of task importance for six occupations into three clusters.
3. Relating to use of softwares, tools for data base management, object oriented softwares, and software packages were mostly used by the programmers.
4. No significant difference between working conditions and occupational differences was noted.
5. Programmers in R & D organizations preferred alone assignment and those in computer training institutes preferred team assignment more than other groups.
6. No significant difference between occurrence of occupational disorder and occupational classification was found. The common occupational disorders were spondilosis, arthritis, sleep disturbance.
7. There was latent relationship among programming tasks. Four factors were extracted using factor analysis. They were program writing and testing, human relations, data analysis and user satisfaction.
8. Programmers in computer training institutes gave more importance on data analysis than other occupations. No significant difference between occupational classification and three task factors (program writing and testing, human relations, and user satisfaction) was noted.
9. Disregarding occupational differences, five specific aptitudes namely comprehension, inference, expressional fluency, associational fluency and alternative usage received more importance.
10. With reference to relative importance of aptitudes, the six occupations were classified into three clusters. Programmers in financial institutes were relatively different from other occupations in terms of aptitude importance.
11. The four task factors were significantly related to various aptitudes of the computer programmers. Program writing and testing was related to

memory for commands, copying, syllogistic reasoning, typing speed, verbal relations, quick grasping of similarities, matrices, matching problems, coding speed, finger dexterity, comprehension, memory for spatial position and memory for digits.

12. Human relations was significantly related to matching identical drawings, spatial visualization, musical ability, manipulatory visualization, decoration word opposites, pattern analysis, perception of fixed spatial and geometrical relations, quick grasping of visual details, matching geometrical forms, quick grasping of visual details, finger dexterity memory for spatial position, matrices, color recognition, alternate uses, sketches, arithmetic calculations, matching problems, word fluency, anagram, causal relations and word classification.
13. *Data analysis* was significantly related to coding speed, finger dexterity, number series anagram, pattern analysis, coding speed, sketches, memory for spatial position, matching geometric forms, color recognition, manipulatory visualization, word fluency, matching identical drawings, arithmetic calculation, perception of fixed spatial or geometrical relations, matching problems, decoration, word classification, vocabulary, and musical ability.
14. *User satisfaction* was significantly related spatial visualization, inference, comprehension, finger dexterity, word fluency, coding speed, copying, color recognition and verbal reasoning.

Findings are useful in vocational selection, guidance and job evaluation of the computer programmers.

### FURTHER RESEARCHES

1. Current study ignored cognitive appraisal of programming tasks and aptitudes of data entry operators and system analysts. So further study is needed to determine significant differences in task and aptitude importance among data entry operators, programmers and system analysts. Findings would reveal much information on career planning for the people in computer operating professions.
2. In sampling, some programming occupations were not found. They are multimedia programming, animated programming. So further study is needed in this direction.
3. Study revealed relative importance of programming tasks and aptitudes. Further study is needed to determine the role of organizational learning process in formation of task and aptitude importance.
4. Project was limited to develop computer algorithm for aptitude test battery. Further attempt is to be made to determine the reliability and validity of aptitude test battery.

## REFERENCES

1. Agarwal, K.K. (1977). Prediction of scholastic success in science subjects: A battery of Psychological Test, Indian Journal of Psychology, 52, 95-97.
2. Anastasi, A. (1988). Psychological Testing, 6<sup>th</sup> Edition, NY, McMillan
3. Avtar, H.S.G. (1965). Validation of psychological tests used for the selection of special class railway apprentices. Indian journal of applied psychology, 2, 45-57.
4. Balgurusamy, E. (1991), Programming in BASIC, (third edition), Tata McGraw Hill Co. Ltd., New Delhi.
5. Bisht, A. R. (1982). Stars and isolates among teachers compared on teaching aptitude. Punitiveness and liking by students. Psychological studies, 27, 90-92.
6. Chadha, N.K. (1996), Theory and Practice of Psychometry, New Delhi, New Age International Limited.
7. Chatterjee, S. and Mukherjee, M. (1978). Higher secondary science achievement as related to scientific interest and aptitudes, Indian Journal of Psychology, 53, 125-129.
8. Cronbach, I.J. (1970) Essentials of Psychological testing, 3rd edition, N.Y.:Harper & Row, Publishers.
9. Cronbach, I.J. (1970) Essentials of Psychological testing, 3rd edition, N.Y.:Harper & Row, Publishers.
10. Deb, M. (1968) Abilities important for achieving science course. Psychological Studies, 81, 31-34.
11. Deb, M. (1973). A factor analytical study of revised form of engineering aptitude test and final form of engineering aptitude test. Psychological studies, 81, 31-34.

12. Dutta Roy, D. and Mallik, R. (2000). Ranking General Aptitudes for Success in computer programming. Journal of the Indian Academy Applied Psychology, 26, 1-2, 135-139.
13. Dutta Roy, D. and Mallik, R. (1999). Factor analysis of computer programming tasks, 5<sup>th</sup> International and 36<sup>th</sup> Indian academy of Applied psychology Conference, University of Bangalore, December 27-29, P.7.
14. Dutta Roy, D. (1994). Relative importance of personality factors in discriminating four occupational groups, Indian Journal of Applied Psychology, 31, 1, 34-38.
15. Freeman, F.S. (1963) Theory and practice of Psychological testing, Calcutta: Oxford & IBH Publishing Co.
16. Garrett, H.E. (1979) Statistics in Psychology and Education, Hyderabad: International Book Bureau. International aptitude testing., 4115 Sherbrooke Street West, Suite 100, X 306, Montreal, Quebec, Canada.
17. Garrett, H.E. (1979) Statistics in Psychology and Education, Hyderabad: International Book Bureau.
18. Jain, V.K. (1987), BASIC Computer Programming, Pustak Mahal, Khari Bari, Delhi, IVth Edition.
19. Kothari, C. R. (1998), Research Methodology: Methods, Techniques, Wishwa Prakashan, New Delhi, second Edition, Reprint.
20. Kristof, A.L. (1996) Person-organization fit: An integrative review of its conceptualizations, measurement, and implications. Personnel Psychology, 49, 1-49.



21. Kurkreti, B.R. (1993). Teaching aptitude and successful teachers : A correlational study: Indian Psychological review. 40,7-8.
22. Mehta, M. (1982). Relationship between scientific interest and scientific aptitude among higher secondary boys. Journal of Psychological Researches, 26, 13-16.
23. Mohon, V. and Brar, A. (1988). Working efficiency of technical personnel in relation to their aptitudes, Journal of Indian Academy of Applied Psychology, 14, 23-40.
24. Munn, N.L., Fernald, L.D., Fernald, P.S. and Carmichael, L. (1978) Introduction to Psychology, Oxford & IBH Publishing Co.
25. Naik, R.B. (1973). Office work aptitude test, Indian Psychological Review, 11, 20-24.
26. Perry, K. (1967), Vocational interest and success of Computer programmers, Personnel Psychology, 20, 4, 517-524.
27. Psychometrics Inc. , Southern California, USA.
28. Psychometrics Inc., <http://www.psy-test.com/Baptd.html>
29. Raina, V.K. (1981). Relationship between personality measures and teaching abilities. Journal of Psychological Researches, 25, 111-115.
30. Roy, G. (1996), Basic for Personal Computer, W. Newman & Company Ltd. Calcutta (Third Edition).
31. Schneider, B. (1987). The people make the place, Personnel Psychology, 40, 437-453.
32. Science Research Associates, Inc., 259 East Erie St., Chicago, III, 60624. Psychometrics Inc. , Southern California, USA.

33. Science Research Associates, Inc., 259 East Erie St., Chicago, III, 60624.
34. Sharma, P. and Shukla, P. (1992). Factor analysis of the comprehensive scientific aptitude test. Indian Journal of Psychology, 67, 47-59.
35. Snow, R. E. (1978). Theory and method for research on aptitude process : A Perspectives. Intelligence, 2, 225-78.
36. The International Programming Aptitude Test , <http://pages.netaxis.ca/winrow/>
37. Tiffin, J and McCormick, E.J. (1971) Manufacturing Psychology, New Delhi: Prentice-Hall of India Private Limited
38. Torrance, E.P. (1966) Torrance tests of creativity, Princeton: Personal Press.
39. Walden Personnel Testing Consulting ,  
<http://www.waldentesting.com/wolfe/wspat.htm>

APPENDICES

# Analysis of Computer programming tasks and Aptitudes

## APPENDIX A

### INDEX PROGRAMME

```
REM THIS PROGRAM ACCEPTS INTRODUCTION OF THE SUBJECT, AND
REM PROGRAM OF 12 APTITUDE TESTS.

CLS
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
LOCATE 9, 20: PRINT "INDIAN STATISTICAL INSTITUTE"
LOCATE 10, 22: PRINT "PSYCHOLOGY RESEARCH UNIT"
LOCATE 11, 10: PRINT "TEST FOR ASSESSMENT OF COMPUTER PROGRAMMING
APTITUDE"
SLEEP 3
CLS
LOCATE 5, 20: INPUT "TYPE YOUR NAME ", NAME$
LOCATE 6, 20: INPUT "TYPE YOUR AGE IN NUMERIC FORM ", AGE
LOCATE 7, 20: INPUT "EDUCATIONAL QUALIFICATIONS ", EDU$
LOCATE 8, 20: INPUT "OCCUPATION ", OCCU$
LOCATE 9, 20: INPUT "ADDRESS IN FULL (PL. AVOID ',') ", ADD$
CLS
LOCATE 5, 5: PRINT "THIS APTITUDE TEST BATTERY WILL ASSESS YOUR
ABILITIES IN DIVERSE FIELDS LIKE NUMERICAL REASONING, ABSTRACT
REASONING, MEMORY, CREATIVITY ETC.INSTRUCTIONS WILL BE PROVIDED
AT THE BEGINNING OF EACH SECTION.READ THE INSTRUCTIONS
CAREFULLY.ALWAYS USE UPPER CASE LETTERS WHILE RESPONDING."
LOCATE 10, 5: PRINT " YOUR RESPONSES WILL BE STRICTLY
CONFIDENTIAL.THE DATA WE ARE COLLECTING WILL BE USED ONLY FOR
RESEARCH PURPOSES.BE COMFORTABLE.FEEL FREE TO APPROACH US FOR ANY
HELP."
LOCATE 22, 5: INPUT "DO YOU UNDERSTAND ?", Y
CLS
PRINT #3, "*****APTITUDE TEST DATA*****"
PRINT #3, NAME$, " AGE :"; AGE
PRINT #3, "EDUCATION :"; EDU$; " OCCUPATION : "; OCCU$
PRINT #3, "ADDRESS : "; ADD$
CLOSE #3
100 CLS
LOCATE 8, 19: PRINT " 1. ABSURDITY"
LOCATE 9, 19: PRINT " 2. ANAGRAM"
LOCATE 10, 19: PRINT " 3. CODING"
LOCATE 11, 19: PRINT " 4. COMPREHENSION"
LOCATE 12, 19: PRINT " 5. CREATIVITY"
LOCATE 13, 19: PRINT " 6. INFERENCE"
LOCATE 14, 19: PRINT " 7. MEMORY"
LOCATE 15, 19: PRINT " 8. NUMBER SERIES"
LOCATE 16, 19: PRINT " 9. NUMERICAL REASONING"
LOCATE 17, 19: PRINT " 10. VOCABULARY"
LOCATE 18, 19: PRINT " 11. END"
LOCATE 21, 19: INPUT "SELECT YOUR TEST BY TYPING NUMBER ", NUM
IF NUM = 0 GOTO 11
ON NUM GOTO 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
1 INPUT "SOURCE OF ABSUR.BAS : ", SOFI$
RUN SOFI$
```

## Analysis of Computer programming tasks and Aptitudes

```
GOTO 100
2 INPUT "SOURCE OF ANAGRAM.BAS : ", SOFI$
RUN SOFI$
GOTO 100

3 INPUT "SOURCE OF CODING.BAS : ", SOFI$
RUN SOFI$
GOTO 100
4 INPUT "SOURCE OF COMPRE.BAS : ", SOFI$
RUN SOFI$
GOTO 100
5 INPUT "SOURCE OF CREATE.BAS : ", SOFI$

RUN SOFI$
GOTO 100
6 INPUT "SOURCE OF INFER.BAS : ", SOFI$
RUN SOFI$
GOTO 100
7 INPUT "SOURCE OF MEMO.BAS : ", SOFI$
RUN SOFI$
GOTO 100
8 INPUT "SOURCE OF AUTO_NEW.EXE : ", SOFI$
SHELL SOFI$
GOTO 100
9 INPUT "SOURCE OF AUTO_NUM.EXE : ", SOFI$
SHELL SOFI$
GOTO 100
10 INPUT "SOURCE OF VOCABUL.BAS : ", SOFI$
RUN SOFI$
GOTO 100
11 END
SUB MEM1
END SUB
```

## Analysis of Computer programming tasks and Aptitudes

### ABSURDITY TEST

```
DIM abq$(15), ABQR$(15)
CLS
REM This is a test for measurement of absurdities
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 60: PRINT TIME$
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
COLOR 2, 8, 11
LOCATE 5, 20: INPUT "TYPE YOUR NAME   ", NAME$
LOCATE 6, 20: INPUT "TYPE YOUR AGE IN NUMERIC FORM   ", AGE
LOCATE 7, 20: INPUT "EDUCATIONAL QUALIFICATIONS   ", EDU$
LOCATE 8, 20: INPUT "OCCUPATION   ", OCCU$
LOCATE 9, 20: INPUT "ADDRESS (PL. AVOID ',') ", ADD$
PRINT #3, DATE$, NAME$, "("; AGE; ")"
PRINT #3, EDU$, OCCU$, ADD$
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"
PRINT #3, "***** ABSURDITY TEST *****"
CLS
LOCATE 10, 30: PRINT " ABSURDITY TEST "
COLOR 7, 8, 11
LOCATE 20, 10: PRINT " PLEASE USE UPPER CASE ALPHABET IN RESPONDING"
LOCATE 22, 10: INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", V
SLEEP 2
CLS
GOSUB 10: REM direction
GOSUB 20: REM testing
CLS
LOCATE 15, 20: PRINT "YOUR ABSURDITY TEST SCORE = ", ABSC
INPUT "LOCATION OF INDEX FILE ", FIORI$
RUN FIORI$
10 REM direction
CLS
LOCATE 4, 5: PRINT "Directions:"
LOCATE 6, 5: PRINT "In each of the following questions, four out of
five items are alike in a certain way and so form a group. Find out
the one item that does not belong to the group and press the key
that corresponds to that answer choice ."
LOCATE 11, 4: PRINT " For example"
LOCATE 13, 4: PRINT " a) BD      b) CE      c) DF      d)FG"
LOCATE 15, 4
PRINT " "
PRINT " Here the correct answer is 'd'. Every second letter is taken
to make the pairs. In `d' this relationship does not exist. so you
just press the key 'a' . To answer this question, press the key `d' on
the keyboard. "
INPUT "Do you understand the direction ? ( yes = 1 ) ", a
CLS
IF a <> 1 GOTO 10
RETURN

20 REM absurdity question
FOR I = 1 TO 15
READ abq$(I), ABQR$(I)
```

## Analysis of Computer programming tasks and Aptitudes

```

x1$ = TIME$
LOCATE 10, 2: PRINT I; ". "; abq$(I)
LOCATE 15, 2: INPUT "Type answer ", abans$
x2$ = TIME$
IF abans$ = ABQR$(I) THEN ABSC = ABSC + 1
CLS
GOSUB 1000
NEXT I

DATA "a) aBC  b) BaC  c) abC  d) BCa ", "C"
DATA " a) BATU  b) ZYST  c) NMPQ  d) IJSO", "D"
DATA " a) TSRQ  b) HGFD  c) MLKJ  d) YXWV ", "B"
DATA " a) bcd fgh  b) opq stu  c) abc efg  d) stu xyz ", "D"
DATA " a) PrtK  b) PiuS  c) PoKe  d) PaKe ", "A"
DATA " a) Tin  b) Steel  c) Iron  d) Copper", "B"
DATA " a) Cricket  b) Chess  c) Hockey  d)Football", "B"
DATA " a) May  b) July  c) September  d) January", "C"
DATA " a) Meow  b) Bray  c) Bark  d) Croak ", "D"
DATA " a) China :Beijing  b) England :London  c) India :Mumbai
d) Japan :Tokyo ", "C"
DATA " a) 27  b) 24  c) 36  d) 12", "A"
DATA " a) 121  b) 12321  c) 123421  d) 123454321 ", "C"
DATA " a) 63852  b) 25683  c) 28751  d) 53268", "C"
DATA " a) 11,3,13,17  b) 41,5,3,47  c) 37,14,19,7  d) 67,71,3,5
", "C"
DATA " a) 22,4,5  b) 37,4,9  c) 34,4,8  d) 54,4,13 ", "B"
CLOSE #3
RETURN

CLS
1000 REM "CALCULATING TIME"
BEEP
sec1$ = RIGHT$(x1$, 2)
sec1 = VAL(sec1$)
sec2$ = RIGHT$(x2$, 2)
sec2 = VAL(sec2$)
IF sec2 < sec1 THEN td = (60 - sec1) + sec2
IF sec2 > sec1 THEN td = sec2 - sec1
PRINT #3, I, ABQR$(I), x1$, x2$, td, ABSC
RETURN

```

## Analysis of Computer programming tasks and Aptitudes

### ANAGRAM TEST

```
CLS
REM This is a test for measurement of ability to solve anagrams
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 60: PRINT TIME$
file$ = "c:\windows\desktop\apti.out "
OPEN file$ FOR APPEND AS #3

COLOR 2, 8, 11
LOCATE 5, 20: INPUT "TYPE YOUR NAME   ", NAME$
LOCATE 6, 20: INPUT "TYPE YOUR AGE IN NUMERIC FORM   ", AGE
LOCATE 7, 20: INPUT "EDUCATIONAL QUALIFICATIONS   ", EDU$
LOCATE 8, 20: INPUT "OCCUPATION   ", OCCU$
LOCATE 9, 20: INPUT "ADDRESS (PL. AVOID ',')   ", ADD$
PRINT #3, DATE$, NAME$, "("; AGE; ")"
PRINT #3, EDU$, OCCU$, ADD$
PRINT #3, "***** ANAGRAM TEST *****"
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"

CLS
LOCATE 10, 30: PRINT " ANAGRAM TEST "
COLOR 7, 8, 11
LOCATE 20, 10: PRINT " PLEASE USE LOWER CASE ALPHABET IN RESPONDING"
LOCATE 22, 10: INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", v
SLEEP 2
CLS
SCREEN 0
PLAY "abcabc"
10 LOCATE 2, 30
   LOCATE 4, 5: PRINT "Directions:"
LOCATE 5, 5: PRINT "THERE ARE 15 ITEMS IN THIS SECTION.EACH ITEM IS
FOLLOWED BY SOME LETTERS OF A MEANINGFUL WORD WHICH ARE NOT
PROPERLY ARRANGED.YOU ARE TO ARRANGE THE LETTERSPROPERLY SO THAT IT
WILL CONVEY THE MEANING OF THE WORD. THERE ARE FIVE
ALTERNATIVE ANSWERS NAMEDLY 'a','b','c','d'AND 'e'.CHOOSE THE CORRECT
ALTERNATIVEAND PRESS THE CORRESPONDING KEY."
LOCATE 12, 4: PRINT "For example"
LOCATE 13, 4: PRINT "i)T ii)N iii)R   iv)A   v)i"
LOCATE 14, 4: PRINT "a)v,i,iii,iv,ii   "
LOCATE 15, 4: PRINT "b)ii,v,iii,i,iv   "
LOCATE 16, 4: PRINT "c)i,iii,iv,v,ii   "
LOCATE 17, 4: PRINT "d)iii,i,ii,iv,v   "
LOCATE 18, 4: PRINT "e)i,v,iv,ii,iii"
PRINT " "
PRINT " Here the correct answer is 'c', so you just press the key
'c' "
INPUT "Do you understand the direction ? ( yes = 1 ) ", a
IF a <> 1 GOTO 10
GOSUB 1000

REM TESTING STARTS NOW
CLS
```



## Analysis of Computer programming tasks and Aptitudes

```

NS = 0
NS = NS + 1

x1$ = TIME$
REM Problem 1
COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)A   ii)S   iii)G   iv)R
v)U "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT "a)ii,v,iv,iii,i  "
LOCATE 13, 20: PRINT "b)i,iii,v,iv,ii  "
LOCATE 14, 20: PRINT "c)iv,ii,iii,i,v  "
LOCATE 15, 20: PRINT "d)ii,v,iii,i,iv  "
LOCATE 16, 20: PRINT "e)v,iii,iv,ii,i"
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 2
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i) R  ii)E   iii)G   iv)I
v)T"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)v,iv,iii,ii,i  "
LOCATE 13, 20: PRINT " b)ii,v,iv,iii,i  "
LOCATE 14, 20: PRINT " c)i,iii,v,iv,ii  "
LOCATE 15, 20: PRINT " d)v,iii,iv,ii,i  "
LOCATE 16, 20: PRINT " e)i,iv,iii,ii,v  "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

REM problem 3
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)I   ii)D   iii)L   iv)F
v)E"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)iv,i,iii,ii,v  "
LOCATE 13, 20: PRINT " b)iv,i,v,iii,ii  "
LOCATE 14, 20: PRINT " c)i,iii,v,iv,ii  "
LOCATE 15, 20: PRINT " d)ii,iv,v,iii,i  "
LOCATE 16, 20: PRINT " e)v,iii.iv.ii,i  "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

```

## Analysis of Computer programming tasks and Aptitudes

```

REM problem 4
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)T  ii)A  iii)P  iv)L  v)N
"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a) ii,v,iv,iii,i  "
LOCATE 13, 20: PRINT " b)iv,i,v,iii,ii  "
LOCATE 14, 20: PRINT " c)iii,iv,ii,v,i  "
LOCATE 15, 20: PRINT " d)ii,v,iv,i,iii  "
LOCATE 16, 20: PRINT " e)v,ii,iv,i,iii"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

```

```

REM problem 5
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 19: PRINT " i)M  ii)E  iii)Y  iv)O
v)N"
COLOR 2, 8, 15: LOCATE 12, 19: PRINT " a)v,iii,iv,ii,i  "
LOCATE 13, 20: PRINT "b)ii,v,iv,i,iii  "
LOCATE 14, 20: PRINT "c)i,iv,v,ii,iii  "
LOCATE 15, 20: PRINT "d)v,i,iii,iv,ii"
LOCATE 16, 20: PRINT "e)iii,i,v,iv,ii"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

```

```

REM problem 6
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)C  ii)E  iii)I  iv)P  v)L
vi)N"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)ii,i,iv,v,iii,vi  "
LOCATE 13, 20: PRINT " b)v,i,iv,iii,ii,vi  "
LOCATE 14, 20: PRINT " c)i,iv,vi,iii,ii,v  "
LOCATE 15, 20: PRINT " d)ii,vi,iii,i,v,iv  "
LOCATE 16, 20: PRINT " e)iv,ii,vi,i,iii,v  "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

```

## Analysis of Computer programming tasks and Aptitudes

```

REM problem 7
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)M   ii)U   iii)N   iv)R
v)E   vi)B"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)v,i,iv,iii,ii,vi  "
LOCATE 13, 20: PRINT " b)iii,ii,i,vi,v,iv  "
LOCATE 14, 20: PRINT " c)ii,vi,iii,i,iv,v  "
LOCATE 15, 20: PRINT " d)iii,ii,i,vi,v,iv  "
LOCATE 16, 20: PRINT " e)vi,i,iii,v,iv,ii"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 8
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)W   ii)E   iii)L   iv)R
v)O   vi)F"
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)vi,iii,v,i,ii,iv  "
LOCATE 13, 20: PRINT " b)i,v,ii,iv, vi, v  "
LOCATE 14, 20: PRINT " c)vi,i,iii,v, iv,ii  "
LOCATE 15, 20: PRINT " d)i,iv, vi,iii,ii,v  "
LOCATE 16, 20: PRINT " e)ii,vi,iii,i,iv, v"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

REM problem 9
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)T   ii)E   iii)M   iv)R   v)A
vi)K  "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)i,v,ii,iv,vi,iii  "
LOCATE 13, 20: PRINT " b)iii,v,iv.vi.ii,i  "
LOCATE 14, 20: PRINT " c)ii,iv,v,iii,i,vi  "
LOCATE 15, 20: PRINT " d)vi,v,iii,iv,i,ii  "
LOCATE 16, 20: PRINT " e)iii,iv,vi,ii,v,i  "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 10
NS = NS + 1
x1$ = TIME$

```

## Analysis of Computer programming tasks and Aptitudes

```

COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)N ii)E iii)T iv)U v)G
vi)O "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)vi,i,iii,v,iv,ii "
LOCATE 13, 20: PRINT " b)vi,iii,v,i,ii,iv "
LOCATE 14, 20: PRINT " c)iii,v,iv,vi,ii,i"
LOCATE 15, 20: PRINT " d)iii,vi,i,v,iv,ii "

```

```

LOCATE 16, 20: PRINT " e)i,ii,v,iv,vi,iii "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

```

```

REM problem 11
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)T ii)O iii)N iv)U v)R
vi)C vii)Y "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a) v,i,iv,iii,vii,ii,vi "
LOCATE 13, 20: PRINT " b) vii,i,iii,iv,ii,vi,v "
LOCATE 14, 20: PRINT " c)ii,i,iv,vii,iii,v, vi "
LOCATE 15, 20: PRINT " d)i,v,vii,ii,iv, vi,iii "
LOCATE 16, 20: PRINT " e)vi,ii,iv,iii, i,v,vii"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

```

```

REM problem 12
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT " i)R ii)O iii)W iv)N v)A
vi)I vii)B "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a) vi,i,vii,iii,v,iv,ii "
LOCATE 13, 20: PRINT " b)i,v,vi,iv,vii,ii,iii "
LOCATE 14, 20: PRINT " c)i,v,vii,iv,iii,ii,vi "
LOCATE 15, 20: PRINT " d)ii,i,iv,iii,vi,v,vii "
LOCATE 16, 20: PRINT " e)iii,vi,ii,iv,i,vii,v "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

```

```

REM problem 13
NS = NS + 1
x1$ = TIME$

```

## Analysis of Computer programming tasks and Aptitudes

```

COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)L ii)A iii)Q iv)I v)T
vi)U vii)Y "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)vi,ii,iii,iv,i,v,vii "
LOCATE 13, 20: PRINT " b)ii,vi,iii,vii,i,v,iv "
LOCATE 14, 20: PRINT " c)iv,i,iii,ii,vii,v,vi "
LOCATE 15, 20: PRINT " d)iii,vi,ii,i,iv,v,vii "
LOCATE 16, 20: PRINT " e)vi,iii,v,iv,i,ii,vii"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$

x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 14
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)K ii)E iii)N iv)T v)I
vii)C viii)H "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a) i,v,iv,vi,vii,ii,iii "
LOCATE 13, 20: PRINT " b)iii,vi,ii,i,iv,v,vii"
LOCATE 14, 20: PRINT " c)vi,ii,iv,iii,i,v,vii "
LOCATE 15, 20: PRINT " d)i,iv,vi,iii,vii,ii,v "
LOCATE 16, 20: PRINT " e)vi,ii,v,i,vii,iv,iii "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

REM problem 15
NS = NS + 1
x1$ = TIME$
COLOR 14, 9, 4: LOCATE 10, 20: PRINT "i)R ii)A iii)V iv)S v)T
vi)E vii)N "
COLOR 2, 8, 15: LOCATE 12, 20: PRINT " a)i,v,vii,ii,iv,vi,iii "
LOCATE 13, 20: PRINT " b)i,v,vi,iv,vii,ii,iii "
LOCATE 14, 20: PRINT " c)iv,vi,i,iii,ii,vii,v "
LOCATE 15, 20: PRINT " d)vii,i,v,iii,ii,vi,iv "
LOCATE 16, 20: PRINT " e)i,iv,vii,vi,iii,ii,v"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

LOCATE 5, 20: PRINT "total score =", score
LOCATE 20, 30: PRINT "THANK YOU"
GOTO 2000

CLS
1000 REM "CALCULATING TIME"
BEEP
sec1$ = RIGHT$(x1$, 2)

```

## Analysis of Computer programming tasks and Aptitudes

```
sec1 = VAL(sec1$)

sec2$ = RIGHT$(x2$, 2)
sec2 = VAL(sec2$)
IF sec2 < sec1 THEN td = (60 - sec1) + sec2
IF sec2 > sec1 THEN td = sec2 - sec1
PRINT #3, NS, R$, x1$, x2$, td, score
RETURN
2000 CLOSE #3
INPUT "location of index file", fiori$
RUN fiori$
```

## Analysis of Computer programming tasks and Aptitudes

### CODING AND DECODING

```
CLS
REM This is a test for measurement of coding and decoding aptitudes
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 70: PRINT TIME$

LOCATE 4, 10: INPUT "TYPE YOUR NAME"           ", NAME$
LOCATE 5, 10: INPUT "TYPE YOUR AGE IN NUMERIC FORM" ", AGE
LOCATE 6, 10: INPUT "EDUCATIONAL QUALIFICATIONS"   ", EDU$
LOCATE 7, 10: INPUT "OCCUPATION"                   ", OCCU$
LOCATE 8, 10: INPUT "ADDRESS"                       ", ADD$
PRINT #3, DATE$, NAME$, AGE, EDU$, OCCU$, ADD$
PRINT #3, "***** coding & decoding test
*****"
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"

CLS
LOCATE 10, 20
COLOR 2, 8, 11
PRINT " CODING AND DECODING TEST "
LOCATE 20, 10
PRINT " PLEASE USE UPPER CASE ALPHABET IN RESPONDING"
PRINT " "
PRINT " "
INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", V
SLEEP 2
10 CLS
COLOR 7, 8
SCREEN 0
PLAY "abcabc"
PRINT "Directions:"
PRINT " "
COLOR 2, 8, 11
PRINT "This test includes 12 items followed by combination of digits
(i.e,i,ii,iii,and iv) and letters(i.e,A, B, C, and D). You are to
find out the right combinations from the alternatives namely a, b,
c, and d, considering the similarity between the combinations of
letters and digits respectively"
PRINT " "
PRINT " For example"
LOCATE 10, 20: PRINT " 1 = C  4 = E  6 = K  9 = A "
LOCATE 11, 20: PRINT " i) 46 ii) 19 iii) 61 iv) 94"
LOCATE 12, 20: PRINT " A) AE B)EC C)AK D)CA "
LOCATE 15, 20: PRINT " a) iB & iiC b)iiD & ivA c)iiiC & ivB d)iiD
& iiiA "
LOCATE 18, 1: PRINT " Here the correct answer is b, so you just
press the key b"
COLOR 14, 9, 4
LOCATE 20, 1: INPUT "Do you understand the direction ? ( yes = 1 )
", A
IF A <> 1 GOTO 10
```

## Analysis of Computer programming tasks and Aptitudes

```
20 REM TESTING STARTS NOW
CLS
NS = 0
NS = NS + 1
x1$ = TIME$
REM Problem 1
LOCATE 5, 20: PRINT " 1 = G   6 = L   3 = V   4 = A   8 = F  "
LOCATE 6, 20: PRINT " i) 63  ii) 41  iii) 86  iv) 18"
LOCATE 7, 20: PRINT " A) FL  B)GF  C)AG  D)LA  "
LOCATE 9, 20: PRINT "a) iD & iiiA  b)iiB & iC  c)ivA & iiiB  d)iiiA
& ivB  "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 2
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=C   3=F   5=H   7=N   2=P   6=S   4=E   "
LOCATE 6, 20: PRINT " i) 26  ii) 47  iii) 15  iv) 73"
LOCATE 7, 20: PRINT " A) HN  B)CF  C)PS  D)EN"
LOCATE 9, 20: PRINT " a) iA & iiiD  b)iiiB & iiD  c)iC & iiD  d)iD &
ivB  "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

REM problem 3
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=F   5=B   9=D       2=P   6=W   3=X   7=L   4=Z
8=U   "
LOCATE 6, 20: PRINT " i) 45  ii) 38  iii) 91  iv) 82"
LOCATE 7, 20: PRINT " A) XU  B)UP  C)DB  D)ZB"
LOCATE 9, 20: PRINT " a) iB & iiiD  b) iiD & iiiA  c)iiB & ivC
d)iiD & iiC  "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 4
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=K   5=S   8=P   2=T   3=R   "
LOCATE 6, 20: PRINT " i) 312  ii) 538  iii) 825  iv) 128"
LOCATE 7, 20: PRINT " A) RSP  B)KST  C)RKT  D)PTS  "
```



## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 9, 20: PRINT " a) iC & ivB b)iiD & iiiA c)iC & iiiD d)iiid
& ia "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 5
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=B 6=I 3=G 8=D 5=H 7=C 2=E"
LOCATE 6, 20: PRINT " i) 568 ii) 762 iii)672 iv)812"
LOCATE 7, 20: PRINT " A) ICE B)BGE C)HID D)DBG"
LOCATE 9, 20: PRINT " a) iC & iiB b)iiB & iiiC c)iC & iiiA d)ivB
&iC"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 6
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 2=D 6=G 3=L 9=P 7=K 4=F 1=M 5=J 8=N"
LOCATE 6, 20: PRINT " i) 273 ii)617 iii)496 iv)546"
LOCATE 7, 20: PRINT " A) JFM B)GMK C)NKD D)FPG"
LOCATE 9, 20: PRINT " a) iA & iiD b)iC & iiiD c)ivD & iiiA d)iiB &
iiid"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 7
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT "1=S 5=Y 2=C 9=P 3=G 4=N "
LOCATE 6, 20: PRINT "i) 2541 ii)3415 iii)5142 iv)3541"
LOCATE 7, 20: PRINT "A) GYNS B)CYNS C)NGYC D)GSNC"
LOCATE 9, 20: PRINT "a) iB & ivA b)iC & iiB c)iB & iiiD d)iiiB &
ivC"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 8
NS = NS + 1
x1$ = TIME$
```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 5, 20: PRINT "3=B 8=D 6=F 9=K 5=X 2=E 7=S"
LOCATE 6, 20: PRINT "i) 3278 ii)8695 iii)5965 iv)7896"
LOCATE 7, 20: PRINT "A) KAFX B)DFKX C)BESD D)BXSE"
LOCATE 9, 20: PRINT "a) iC & ivB b)iC & iiB c)iiiA & ivC d)iiB &
iiiA"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 9
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=E 5=J 2=W 4=A 3=Y 7=B 9=P 6=Q 8=M "
LOCATE 6, 20: PRINT " i) 9273 ii) 8546 iii) 9861 iv) 7325 "
LOCATE 7, 20: PRINT " A) QWAY B)BYWJ C)PWBY D)MPJE"
LOCATE 9, 20: PRINT " a) ivA & iiiB b)iD & iiB c)iA & iiC d)iC &
ivB "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 10
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 1=S 5=Y 2=C 4=N 3=G "
LOCATE 6, 20: PRINT " i) 25413 ii) 34152 iii)51423 iv) 35421 "
LOCATE 7, 20: PRINT " A) CYNCG B)GYCSN C)NSYCG D)YSNCG"
LOCATE 9, 20: PRINT " a) ivA & iiC b)iB & iiiC c)ivC & iiA d)iA &
iiid "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 11
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 5=A 7=C 1=P 6=Q 3=L 4=T 9=V "
LOCATE 6, 20: PRINT " i)43716 ii) 69574 iii)93146 iv) 13457 "
LOCATE 7, 20: PRINT " A)QVACT B)TLCPQ C)PVQAT D)ALTQP"
LOCATE 9, 20: PRINT " a) iB & iiA b)iC & ivD c)iB & iiid d)iiD &
iB"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
```

## Analysis of Computer programming tasks and Aptitudes

```
REM problem 12
NS = NS + 1
x1$ = TIME$
LOCATE 5, 20: PRINT " 2=K  7=B  5=M  3=P   8=T  6=C  1=D   4=V  9=Y
"
LOCATE 6, 20: PRINT " i)16537   ii) 94273   iii)68349   iv) 12354   "
LOCATE 7, 20: PRINT " A)TCDPB  B)DKPMV  C)YVKBP  D)CTPBK"
LOCATE 9, 20: PRINT " a) iC & ivB  b)iiC & ivB  c)iiiC & iD  d)iiB &
ivD"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
GOTO 2000

REM -----END-----
CLS
1000 REM "CALCULATING TIME"
BEEP
sec1$ = RIGHT$(x1$, 2)
sec1 = VAL(sec1$)
sec2$ = RIGHT$(x2$, 2)
sec2 = VAL(sec2$)
IF sec2 < sec1 THEN td = (60 - sec1) + sec2
IF sec2 > sec1 THEN td = sec2 - sec1
PRINT #3, NS, R$, x1$, x2$, td, score
RETURN
2000 CLOSE #3
INPUT "location of index file", fiori$
RUN fiori$
```

## Analysis of Computer programming tasks and Aptitudes

### COMPREHENSION TEST

```
CLS
DIM q$(25), a$(25), AW$(25), CA$(35)
REM This is a test for measurement of ability to comprehend.
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
COLOR 2, 8, 14
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 70: PRINT TIME$

CLS
LOCATE 10, 20
COLOR 2, 8, 11
PRINT " COMPREHENSION TEST  "
SLEEP 2
REM "-----TEST STARTING-----"
GOSUB 15: REM passage reading
GOSUB 20: REM Direction 1
GOSUB 30: REM Direction 2
TSCORE = CSCORE + SCORE1 + SCORE2
PRINT #3, " FIRST PROBLEM SCORE =", CSCORE
PRINT #3, "SECOND PROBLEM SCORE =", SCORE1
PRINT #3, "THIRD PROBLEM SCORE=", SCORE2
PRINT #3, "TOTAL SCORE=", TSCORE
CLOSE #3
INPUT "LOCATION OF INDEX FILE ", FIORI$
RUN FIORI$
REM "-----ENDING THE PROGRAM-----"

15 REM passage reading
PRINT #3, "COMPREHENSION FROM PASSAGE"
FOR I = 1 TO 5
READ CA$(I)
DATA "A","C", "D", "A","C"
NEXT I
CLS
COLOR 14, 9, 4
LOCATE 10, 25: PRINT "A passage along with a set of questions will
be shown to you.Read the passage thoroughly and answer the
questions that follow.Indicate your answer choice by pressing the
appropriate key on the keyboard."
SLEEP 5
LOCATE 1, 30: PRINT "THE PASSAGE"
COLOR 14, 9, 4: LOCATE 5, 2: PRINT " That year 1930,was full of
dramatic situations and inspiring happenings:what surprised most,
was the amazing power of Gandhiji,to inspire and enthuse a whole
people.There was something almost hypnotic about it,and we
remembered the words,used by Gokhale about him,-how he had the power
of making heroes out of clay. Peaceful civil disobidience as a
technique of action for achieving great natinal ends,seemed to have
justified itself,and a quiet confidence grew in the country shared
by friend and opponent alike,that we were marching towards victory.A
strange excitement filled those, who were active in the movement and
some of these even crept inside the gaol.`Swaraj is coming',said
```

## Analysis of Computer programming tasks and Aptitudes

the ordinary convicts and they waited impatiently for it, in the selfish hope that it might do them good.The

warders,coming in contact with the gossip of the bazzars,also expected that Swaraj was near.The petty gaol official grew a little more nervous."

PRINT

INPUT "Have you finished reading ? ", a

CLS

X1\$ = TIME\$

COLOR 14, 9, 4: LOCATE 2, 1: PRINT " That year 1930,was full of dramatic situations and inspiring happenings:what surprised most, was the amazing power of Gandhiji,to inspire and enthuse a whole people.There was something almost hypnotic about it,and we remembered the words,used by Gokhale about him,-how he had the power of making heroes out of clay. Peaceful civil disobidience as a technique of action for achieving great natinal ends,seemed to have justified itself,and a quiet confidence grew in the country shared by friend and opponent alike,that we were marching towards victory.A strange excitement filled those, who were active in the movement and some of these even crept inside the gaol.`Swaraj is coming',said the ordinary convicts and they waited impatiently for it, in the selfish hope that it might do them good.The warders,coming in contact with the gossip of the bazzars,also expected that Swaraj was near.The petty gaol official grew a little more nervous."

LOCATE 16, 2: PRINT " 1. The year 1930 was full of dramatic situation because "

LOCATE 17, 2: PRINT " a. Peaceful disobidience as a technique of action seemed to have justified itself."

LOCATE 20, 2: PRINT " b. Gandhiji inspired people."

LOCATE 21, 2: PRINT " c. The British government was nervous."

LOCATE 22, 2: PRINT " d. The masses become conscious of their civil rights."

LOCATE 24, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS\$

X2\$ = TIME\$

IF ANS\$ = CA\$(1) THEN CSCORE = CSCORE + 1

PRINT #3, "1. INITIAL ="; X1\$, "FINAL ="; X2\$, "SCORE ="; CSCORE

CLS

X1\$ = TIME\$

COLOR 14, 9, 4: LOCATE 2, 1: PRINT " That year 1930,was full of dramatic situations and inspiring happenings:what surprised most, was the amazing power of Gandhiji,to inspire and enthuse a whole people.There was something almost hypnotic about it,and we remembered the words,used by Gokhale about him,-how he had the power of making heroes out of clay.Peaceful civil disobidience as a technique of action for achieving great natinal ends,seemed to have justified itself,and a quiet confidence grew in the country shared by friend and opponent alike,that we were marching towards victory.A strange excitement filled those, who were active in the movement and some of these even crept inside the gaol.`Swaraj is coming',said the ordinary convicts and they waited impatiently for it, in the selfish hope that it might do them good.The warders,coming in contact with the gossip of the bazzars,also expected that Swaraj was near.The petty gaol official grew a little more nervous."

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 16, 5: PRINT "    2. What did one and all think at the time?"
LOCATE 18, 5: PRINT "    a. The imperial power was collapsing."
LOCATE 19, 5: PRINT "    b. India would become free very shortly."
LOCATE 20, 5: PRINT "    c. Swaraj was in sight."
LOCATE 21, 5: PRINT "    d. Gandhiji became the leader of the
masses."
LOCATE 25, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
X2$ = TIME$
IF ANS$ = CA$(2) THEN CSCORE = CSCORE + 1
PRINT #3, "2.    INITIAL ="; X1$, "FINAL ="; X2$, "SCORE ="; CSCORE

CLS
X1$ = TIME$
COLOR 14, 9, 4: LOCATE 2, 1: PRINT " That year 1930,was full of
dramatic situations and inspiring happenings:what surprised most,
was the amazing power of Gandhiji,to inspire and enthuse a whole
people.There was something almost hypnotic about it,and we
remembered the words,used by Gokhale about him,-how he had the power
of making heroes out of clay.Peaceful civil disobidience as a
technique of action for achieving great natinal ends,seemed to have
justified itself,and a quiet confidence grew in the country shared
by friend and opponent alike,that we were marching towards victory.A
strange excitement filled those, who were active in the movement and
some of these even crept inside the gaol.`Swaraj is coming',said the
ordinary convicts and they waited impatiently for it, in the selfish
hope that it might do them good.The warders,coming in contact with
the gossip of the bazzars,also expected that Swaraj was near.The
petty gaol official grew a little more nervous."
LOCATE 16, 5: PRINT "    3. How did the participants in the movement
feel?"
LOCATE 18, 5: PRINT "    a. They were hesitant."
LOCATE 19, 5: PRINT "    b. They were reluctant."
LOCATE 20, 5: PRINT "    c.They were compelled."
LOCATE 21, 5: PRINT "    d.They were quite excited."
LOCATE 25, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
X2$ = TIME$
IF ANS$ = CA$(3) THEN CSCORE = CSCORE + 1
PRINT #3, "3.    INITIAL ="; X1$, "FINAL ="; X2$, "SCORE ="; CSCORE

CLS
X1$ = TIME$
COLOR 14, 9, 4: LOCATE 2, 1: PRINT " That year 1930,was full of
dramatic situations and inspiring happenings:what surprised most,
was the amazing power of Gandhiji,to inspire and enthuse a whole
people.There was something almost hypnotic about it,and we
remembered the words,used by Gokhale about him,-how he had the power
of making heroes out of clay.Peaceful civil disobidience as a
technique of action for achieving great natinal ends,seemed to have
justified itself,and a quiet confidence grew in the country shared
by friend and opponent alike,that we were marching towards victory.A
strange excitement filled those, who were active in the movement and
some of these even crept inside the gaol.`Swaraj is coming',said the
ordinary convicts and they waited impatiently for it, in the selfish
hope that it might do them good.The warders,coming in contact with
the gossip of the bazzars,also expected that Swaraj was near.The
petty gaol official grew a little more nervous."
LOCATE 16, 5: PRINT "    4. What made the situation dynamic?"
```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 18, 5: PRINT "    a. The hynotic power of Gandhiji."
LOCATE 19, 5: PRINT "    b. The weakness of the government."
LOCATE 20, 5: PRINT "    c. Gandhiji's personality."
LOCATE 21, 5: PRINT "    d. The great innate potential of the
nation."
LOCATE 25, 5: LOCATE 25, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
X2$ = TIME$
IF ANS$ = CA$(4) THEN CSCORE = CSCORE + 1
PRINT #3, "4.    INITIAL ="; X1$, "FINAL ="; X2$, "SCORE ="; CSCORE
```

```
CLS
```

```
X1$ = TIME$
```

```
COLOR 14, 9, 4: LOCATE 2, 1: PRINT " That year 1930,was full of
dramatic situations and inspiring happenings:what surprised most,
was the amazing power of Gandhiji,to inspire and enthuse a whole
people.There was something almost hypnotic about it,and we
remembered the words,used by Gokhale about him,-how he had the power
of making heroes out of clay.Peaceful civil disobidience as a
technique of action for achieving great natinal ends,seemed to have
justified itself,and a quiet confidence grew in the country shared
by friend and opponent alike,that we were marching towards victory.A
strange excitement filled those, who were active in the movement and
some of these even crept inside the gaol.`Swaraj is coming',said the
ordinary convicts and they waited impatiently for it, in the selfish
hope that it might do them good.The warders,coming in contact with
the gossip of the bazzars,also expected that Swaraj was near.The
petty gaol official grew a little more nervous."
```

```
LOCATE 16, 5: PRINT "    5. What made the jail officials feel
nervous?"
```

```
LOCATE 18, 5: PRINT "    a. They were afraid of losing their jobs."
```

```
LOCATE 19, 5: PRINT "    b. They wee by nature servile."
```

```
LOCATE 20, 5: PRINT "    c. They were afraid of reprisal in free
India."
```

```
LOCATE 21, 5: PRINT "    d. They thought that political prisoners
would become"
```

```
LOCATE 22, 5: PRINT "        their masters very soon."
```

```
LOCATE 25, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
```

```
X2$ = TIME$
```

```
IF ANS$ = CA$(5) THEN CSCORE = CSCORE + 1
```

```
PRINT #3, "5.    INITIAL ="; X1$, "FINAL ="; X2$, "SCORE ="; CSCORE
```

```
PRINT "TOTAL CONPREHENSION PASSAGE SCORE ", CSCORE
```

```
RETURN
```

```
20 REM COMPREHENSION
```

```
PRINT #3, "*****REASONING TEST*****"
```

```
CLS
```

```
LOCATE 5, 30: PRINT " DIRECTION"
```

```
LOCATE 9, 5: PRINT "In the following section there are five
sentences.Next to each sentence there are five answers.Theses are
marked by A,B,C and D.Only one of these suggested answers is
right.You are to pick out the one correct answer for each
sentence.Then press the corresponding key on the keyboard."
```

```
PRINT "For example,"
```

```
PRINT "Keyboard is related to computer as reciever is related
to....."
```

```
PRINT " a)Cord b) switchboard c) Telephone d) Plug "
```

## Analysis of Computer programming tasks and Aptitudes

```

PRINT " in this example, the correct answer is telephone, so you
press the key 'C'."
LOCATE 22, 5: INPUT "DO YOU UNDERSTAND ? ", Y

FOR I = 1 TO 5
CLS
READ q$(I), a$(I), CA$(I)
X1$ = TIME$
LOCATE 9, 25: PRINT q$(I)
PRINT
PRINT a$(I)
LOCATE 15, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
X2$ = TIME$
IF ANS$ = CA$(I) THEN SCORE = SCORE + 1
GOSUB 1000
CLS
NEXT I

DATA "Venus is related to solar system in the same way as student
is related to_____", "A. Class B. Room C. Book D. Teacher","A"
DATA "Table is related to wood in the same way as coat is related
to_____", " A. Tie B. Cloth C.Suit D.Shirt ", "B"
DATA "ABCD is related to HGFE in the same way as KLMN is related
to_____", "A. RQPO B. LMNO C. UVWX D.WXYZ ", "A"
DATA "Hunger is related to desire in the same way as thirst is
related to _____", "A. Understanding B. Need C.Tolerance
D.Necessity ", "B"
DATA "4 is related to 16 in the same way as 5 is related
to_____", "A. 20 B. 25 C. 30 D. 40", "B"
CLS
SCORE1 = SCORE
RETURN

30 REM COMPREHENSION
PRINT #3, " **          SECOND REASONING TEST          ** "
SCORE = 0
REM COLOR 2, 8, 11
LOCATE 2, 30: PRINT " DIRECTION"
LOCATE 5, 5: PRINT "In the following section there are fifteen
sentences.In each of these sentences,the first and the last word is
omitted , and a blank space is provided in its  place.You are to
pick up words which will fill the blanks so that the sentences
become meaningful.You are to choose your words to fill the blanks
from a list of words given with every question.An example will make
it clear.
"
LOCATE 11, 2: PRINT "For example, ----- is to school as employee is
to -----"
LOCATE 13, 5: PRINT " 1) Table  2) Student  3) Book  4) Celing  "
LOCATE 14, 5: PRINT " a) Factory b) Boss c) Manager d) Wall"
LOCATE 16, 5: PRINT " The correct word for the first blank is
student (word no. 2)  and the correct  word for the second blank is
factory ('a')."
LOCATE 19, 4: PRINT " The sentence will then read as student is to
school as employee is to factory"
LOCATE 21, 1: PRINT " The correct answer is 2a. So you press the
keys '2' and 'a'"
LOCATE 24, 5: INPUT "DO YOU UNDERSTAND ?", Y

```



## Analysis of Computer programming tasks and Aptitudes

```

FOR I = 1 TO 15
CLS
READ q$(I), a$(I), AW$(I), CA$(I)
X1$ = TIME$
LOCATE 9, 5: PRINT q$(I)
LOCATE 11, 5: PRINT a$(I)
LOCATE 13, 5: PRINT AW$(I)
LOCATE 15, 5: INPUT "CHOOSE THE CORRECT ANSWER ", ANS$
X2$ = TIME$
IF ANS$ = CA$(I) THEN SCORE = SCORE + 1
SCORE2 = SCORE
GOSUB 1000
NEXT I

DATA "_____ is to beginning as stop is to _____", "1. Start 2.Run
3.Close 4. Step ", "A. End B. Middle C. Inter D. Between
", "1A"

DATA " _____ is to education as treatment is to _____ ", "1. Nurse
2. Teacher 3. School 4.Study ", "A. Policy B. Playing C.
Doctor D. Therapy ", "2C"

DATA " _____ is to furniture as shirt is to _____" , "1. Metal
2. Paper 3. Room 4. Wood ", "A. Cloth B. Feather C. Water
D. Tie ", "4A"

DATA " _____ is to desert as island is to _____" , "1. Peninsula
2.River 3.Oasis 4. Step ", "A. Tower B. Temple C. Ocean
D. Sky", "1A"

DATA " _____ is to coast as river is to _____" , "1. Tea 2. Water
3. Sand 4. Sea ", "A. Boat B. Bank C. Tree D. Sailor ", "4B"

DATA " _____ is to banana as mammal is to _____", "1. Cell 2.
Fruit 3. Art 4. Flower ", "A. Cow B. Snake C. Fish D.
Sparrow ", "2A"

DATA " _____ is to success as initiative is to _____", "1.
Promotion 2.Hard work 3.Intelligence 4.Colorful ", "A.
Encourage B. Punishment C. Development D. Excellent ", "1C"

DATA " _____ is to bright as dull is to _____", " 1. Intelligent
2. Poor 3. Good 4. Scientist ", "A. Non-impressive B.
Background C.Rough D. Calm ", "1A"

DATA " _____ is to conclusion as practice is to _____", "1.
Criticise 2. Manipulation 3. Argument 4. Curing ", "A.
Perfection B. Assumption C. Guess D. Originality ", "3A"

DATA " _____ is to rain as coconut is to _____", "1. Water 2.
Tree 3. Boat 4. Cloud ", "A. Cooking B. Room C. Book D.
Oil ", "4D"

```

## Analysis of Computer programming tasks and Aptitudes

DATA "\_\_\_\_\_ is to fish as air is to \_\_\_\_\_" , "1. Water 2. Food  
3. Hilsa 4. Smoke ", "A. Animal B. Bird C. Man D. Aeroplane  
", "1A"

DATA "\_\_\_\_\_ is to soothe as irritate is to \_\_\_\_\_", "1. Hurt 2.  
Disappoint 3. Pleasure 4. Cheerful ", "A. Trouble B. Please  
C. Respect D. Smile ", "1B"

DATA "\_\_\_\_\_ is to optimist as dark is to \_\_\_\_\_" , "1. Day 2.  
Bright 3. Night 4. Color ", "A. Moon B. Black C. Pessimist  
D. Colorless ", "2C"

DATA "\_\_\_\_\_ is to green as red is to \_\_\_\_\_" , "1. Go 2.  
Continue 3. Come 4. Pull ", "A. Danger B. Symbol C. Stop D.  
Bring ", "1C"

DATA "\_\_\_\_\_ is to sedative as grief is to \_\_\_\_\_", "1. Comfort  
2. Trance 3. Pain 4. Sleep ", "A. Stimulant B. Narcotics C.  
Consolation D. Extradition ", "3D"

CLS  
RETURN  
CLS

```
1000 REM "CALCULATING TIME"  
BEEP  
sec1$ = RIGHT$(X1$, 2)  
sec1 = VAL(sec1$)  
sec2$ = RIGHT$(X2$, 2)  
sec2 = VAL(sec2$)  
IF sec2 < sec1 THEN td = (60 - sec1) + sec2  
IF sec2 > sec1 THEN td = sec2 - sec1  
PRINT #3, "TRIAL: "; I; " INITIAL: "; X1$; " FINAL: "; X2$; " TIME  
DIFF: "; td; " RESPONSE: "; ANS$; " SCORE: "; SCORE  
RETURN  
2000 CLOSE #3  
INPUT "LOCATION OF INDEX FILE", FIORI$  
RUN FIORI$
```

## Analysis of Computer programming tasks and Aptitudes

### CREATIVITY TEST

```
CLS
REM This is a test for measurement of crativity.
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
COLOR 2, 8, 14
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 70: PRINT TIME$
```

```
CLS
LOCATE 10, 20
COLOR 2, 8, 11
PRINT " CREATIVITY TEST  "
SLEEP 2
```

```
GOSUB 15: REM WORD FLUENCY
GOSUB 20: REM IDEATIONAL FLUENCY
GOSUB 25: REM ASSOCIATIONAL FLUENCY
GOSUB 30: REM ASSOCIATIONAL FLUENCY II
GOSUB 35: REM ALTERNATE USAGE
GOSUB 40: REM EXPRESSIONAL FLUENCY
GOSUB 45: REM ORIGINALITY
GOSUB 50: REM ORIGINALITY I
GOSUB 55: REM ORIGINALITY II
PRINT "thank you "
INPUT "LOCATION OF INDEX FILE", FIORI$
RUN FIORI$
```

```
15 REM WORD FLUENCY
COLOR 2, 8, 11
LOCATE 5, 30: PRINT " DIRECTION"
LOCATE 9, 5: PRINT " List insects, fruits, vegetables, floweres and
trees whose names begin"
PRINT "with the letters 'a', 'e' or 'o'. Please insert ',' between
words.You have 2 minutes per item .If you finish writing before
the stipulated time, press enter to go to the next screen."
SLEEP 600
CLS
RETURN
```

```
20 REM IDEATIONAL FLUENCY
COLOR 14, 9, 4
REM PRINT #3, "*****IDEATIONAL FLUENCY RESPONSES*****"
FOR i = 1 TO 6
READ IF$
LOCATE 5, 30: PRINT "DIRECTION"
LOCATE 9, 5: PRINT " Below are given a few questions. The time limit
for each question is 2 minutes.If you finish writing before the
stipulated time,press enter to go on the next screen."
LOCATE 14, 5: PRINT IF$
SLEEP 120
```

## Analysis of Computer programming tasks and Aptitudes

CLS

```
DATA "a) LIST NAME OF THINGS THAT ARE ROUND."
DATA "b) LIST NAME OF THINGS THAT ARE SWEET IN TASTE. "
DATA "c) LIST NAME OF THINGS THAT ARE SOFT AND FLOAT IN AIR. "
DATA "d) LIST NAME OF THINGS THAT ARE SHARP AND USED FOR EVERYDAY
PURPOSES. "
DATA "e) LIST NAME OF THINGS THAT ARE RED, SQUARE AND HOT. "
DATA "f) LIST NAME OF THINGS THAT ARE HARD,BRITTLE AND CAN BE OF
MANY COLOR ."
NEXT i
RETURN
```

25 REM ASSOCIATIONAL FLUENCY

CLS

COLOR 14, 9, 4

REM PRINT #3, "\*\*\*\*\*ASSOCIATIONAL FLUENCY RESPONSES 1 \*\*\*\*\*"

FOR i = 1 TO 5

READ IF\$

LOCATE 5, 30: PRINT "DIRECTION"

LOCATE 9, 5: PRINT " Below are given a number of words.Give  
single,meaninigful verbal responses"

PRINT "against each word to satisfy a relationship.Try to give as  
many responses as "

PRINT "For example,the word 'black' may call for responses such as  
dark,lightless, hopeless etc. The time limit for each item is 2  
minutes.If you finish writing before the stipulated time,press  
enter to go to the next screen."

LOCATE 16, 5: PRINT IF\$

SLEEP 120

CLS

DATA "a) RAIN " , "b) FAIRY " , "c) MONEY " , " d) FRIEND " , "e)  
CHILDHOOD"

NEXT i

RETURN

30 REM ASSOCIATIONAL FLUENCY II

CLS

COLOR 14, 9, 4

REM PRINT #3, "\*\*\*\*\*ASSOCIATIONAL FLUENCY RESPONSES 2 \*\*\*\*\*"

FOR i = 1 TO 5

READ IF\$

LOCATE 5, 30: PRINT "DIRECTION"

LOCATE 9, 5: PRINT "Below are given several pairs of stimulus  
words.In each pair one word indicates one sense and the other a  
quite different sense.Please draw brief word pictures that utilise  
both stimulus words .For example,sweet smell and rough texture can  
make one think of dried rose petals,freshly baked bread,earth after  
the first shower and so on.You have two minutes per item.If you  
finish writing before the stipulated time,press enter to go to the  
next screen."

LOCATE 20, 5: PRINT IF\$

SLEEP 120

CLS

## Analysis of Computer programming tasks and Aptitudes

```
DATA "a) BLUE COLOR AND LOUD SOUND " , "b) SOUR TASTE AND MONOTONOUS  
SOUND " , "c)SILK TEXTURE AND SMELL OF ROTTEN EGGS " , "  
d)FRAGRANCE AND THE GLEAM OF A DIAMOND " , "e)SPICY TASTE AND  
PUNGENT ODOUR"  
NEXT i  
RETURN
```

```
35 REM ALTERNATIVE USES  
CLS  
COLOR 14, 9, 4  
REM PRINT #3, "*****ALTERNATIVE USES *****"  
FOR i = 1 TO 5  
READ IF$  
LOCATE 5, 30: PRINT "DIRECTION"  
PRINT  
LOCATE 9, 5: PRINT "Below are given a number of items. The time  
limit per item is 2 minutes.If you finish writing before the  
stipulated time,press enter to go to the next screen."  
LOCATE 12, 5: PRINT IF$  
SLEEP 120  
CLS
```

```
DATA "A) A NEWSPAPER IS ORDINARILY USED FOR READING.GIVE SOME OTHER  
KINDS OF USES OF A NEWSPAPER."  
DATA "B) A PEN IS ORDINARILY USED FOR WRITING.GIVE SOME OTHER KINDS  
OF USES OF A PEN."  
DATA "C) A SAFETY-PIN IS ORDINARILY USED FOR BINDING.GIVE SOME OTHER  
KINDS OF USES OF A SAFETY-PIN"  
DATA "D) A CUP IS ORDINARILY USED FOR CONTAINING SOMETHING.GIVE SOME  
OTHER KINDS OF USES OF A CUP."  
DATA "E) A MATCHSTICK IS ORDINARILY USED FOR LIGHTING A FIRE.GIVE  
SOME OTHER KINDS OF USES OF A MATCHSTICK.  
NEXT i  
RETURN
```

```
40 REM EXPRESSIONAL FLUENCY  
CLS  
COLOR 14, 9, 4  
REM PRINT #3, "*****EXPRESSIONAL FLUENCY *****"  
FOR i = 1 TO 4  
READ IF$  
LOCATE 5, 30: PRINT "DIRECTION"  
LOCATE 9, 5: PRINT "Below are given a couple of words.Try to use  
these words into as many meaningful sentences as you can.Avoid using  
connectives like `and', `but', `therefore',  
`which', `what', `whom'etc. The time limit for each item is 2  
minutes.If you finish writing before the stipulated time,press  
enter to go to the next screen."  
LOCATE 15, 5: PRINT IF$  
SLEEP 120  
CLS  
DATA "A) STAR, MUSIC "  
DATA "B) FRUIT, HILL "  
DATA "C) COLOR ,COVER, DREAM "
```

## Analysis of Computer programming tasks and Aptitudes

```
DATA "D) TREE, TOWN, GIRL"  
NEXT i  
RETURN
```

```
45 REM ORIGINALITY  
CLS  
COLOR 14, 9, 4  
REM PRINT #3, "***** ORIGINALITY *****"  
FOR i = 1 TO 3  
READ IF$  
LOCATE 5, 30: PRINT "DIRECTION"  
LOCATE 7, 5: PRINT "Below are given the beginnings of some  
sentences. Complete the sentences in as many ways you can think  
of. The time per item is 2 minutes. If you finish writing before the  
stipulated time, press enter to go to the next screen."  
LOCATE 15, 5: PRINT IF$  
SLEEP 120  
CLS  
DATA "A) IT IS AS HOT AS _____ "  
DATA "B) THE MEANING IS AS CLEAR AS _____ "  
DATA "C) THE CAT'S FUR IS AS SOFT AS _____ "  
NEXT i  
RETURN
```

```
50 REM ORIGINALITY 1  
CLS  
REM PRINT #3, "***** ORIGINALITY 1 *****"  
LOCATE 5, 30: PRINT "DIRECTION"  
LOCATE 7, 5: PRINT "Below is given a story which all of us know. Try  
to give an innovative title to"  
PRINT "the story within two minutes."  
PRINT  
COLOR 14, 9, 4: PRINT " A fox was passing by a grapevine which bore  
ripe fruits. He was tempted by the sight of the fruits. He jumped up  
to catch hold of the lowest branch but it was well beyond his reach.  
He tried again and again but failed. Finally, he let off a sigh and  
said - grapes are sour, and went away. "  
SLEEP 120  
RETURN
```

```
55 REM ORIGINALITY III  
CLS  
COLOR 14, 9, 4  
REM PRINT #3, "***** ORIGINALITY III *****"  
LOCATE 5, 30: PRINT "DIRECTION"  
LOCATE 9, 5: PRINT "Suggest some possible ways of making people  
laugh. Suggest as many ways as you can think of. The time limit  
is two minutes."  
SLEEP 120  
RETURN
```

## Analysis of Computer programming tasks and Aptitudes

### INFERENCE TEST

```
CLS
REM This is a test for measurement of INFERENCE
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 70: PRINT TIME$

CLS
LOCATE 10, 20
COLOR 2, 8, 11
PRINT " INFERENCE "
LOCATE 20, 10
PRINT " PLEASE USE UPPER CASE ALPHABETS IN RESPONDING"
LOCATE 23, 10: INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", V
PRINT #3, "*****INFERENCE TEST 1*****"
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"

10 CLS
COLOR 7, 8
SCREEN 0
LOCATE 2, 2: PRINT "Directions:"
PRINT " "
COLOR 2, 8, 11
PRINT "BELOW ARE GIVEN TWO STATEMENTS FOLLOWED BY TWO CONCLUSIONS
NUMBERED (a) AND (b). "
PRINT "YOU HAVE TO TAKE TWO GIVEN STATEMENTS TO BE TRUE EVEN IF THEY
SEEM TO BE AT VARIANCE FROM COMMONLY KNOWN FACTS AND THEN DECIDE
WHICH OF THE GIVEN CONCLUSIONS LOGICALLY FOLLOWS FROM THE
TWO GIVEN STATEMENTS, DISREGARDING COMMONLY KNOWN FACTS."
PRINT
PRINT "GIVE ANSWER "
PRINT " (A) IF ONLY CONCLUSION (a)
FOLLOWS."
PRINT " (B) IF ONLY CONCLUSION (b) FOLLOWS"
PRINT " (C) IF EITHER (a) OR (b) FOLLOWS"
PRINT " (D) IF NEITHER (a) NOR (b) FOLLOWS"
PRINT " (E) IF BOTH (a) AND (b) FOLLOWS.
"
INPUT "DO YOU UNDERSTAND ?", Y

CLS
PRINT "FOR EXAMPLE"
LOCATE 2, 30: PRINT " STATEMENTS"
LOCATE 3, 30: PRINT " i) All pencils are bricks."
LOCATE 4, 30: PRINT "ii) All bricks are bottles."
LOCATE 7, 30: PRINT "CONCLUSIONS"
LOCATE 8, 30: PRINT "a)All pencils are bottles."
LOCATE 9, 30: PRINT "b)All bricks are pencils."
LOCATE 12, 30: PRINT " A B C D E "
LOCATE 14, 30: INPUT "TYPE YOUR ANSWER ", R$
```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 16, 2: PRINT "It follows that since all pencils are bricks
and all bricks are bottles,all pencils must be bottles.But all
bricks may or may not be pencils. Hence, only conclusion (a)
follows. Therefore, the correct answer is (A)."
```

```
PRINT
```

```
LOCATE 20, 2: INPUT " DO YOU UNDERSTAND (IF YES, PRESS KEY 1)  ", A
IF A <> 1 GOTO 10
```

```
CLS
```

```
NS = 0
```

```
NS = NS + 1
```

```
REM problem no 1
```

```
NS = NS + 1
```

```
x1$ = TIME$
```

```
GOSUB 150
```

```
COLOR 2, 8, 11: LOCATE 9, 30: PRINT " STATEMENTS"
```

```
LOCATE 10, 30: PRINT " i) All doors are windows."
```

```
LOCATE 11, 30: PRINT "ii) Some windows are sticks."
```

```
LOCATE 13, 30: PRINT "CONCLUSIONS"
```

```
LOCATE 14, 30: PRINT "a)All doors are sticks "
```

```
LOCATE 15, 30: PRINT "b)Some sticks are doors"
```

```
LOCATE 17, 30: PRINT " A   B   C   D   E   "
```

```
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER  ", R$
```

```
x2$ = TIME$
```

```
IF R$ = "D" THEN score = score + 1
```

```
GOSUB 1000
```

```
CLS
```

```
REM problem no 2
```

```
NS = NS + 1
```

```
x1$ = TIME$
```

```
GOSUB 150
```

```
COLOR 2, 8, 11: LOCATE 9, 30: PRINT " STATEMENTS"
```

```
LOCATE 10, 30: PRINT " i)All Professors are teachers."
```

```
LOCATE 11, 30: PRINT " ii)Some teachers are anxious."
```

```
LOCATE 13, 30: PRINT " CONCLUSIONS"
```

```
LOCATE 14, 30: PRINT " a) Some teachers are Professors "
```

```
LOCATE 15, 30: PRINT " b) All Professors are anxious"
```

```
LOCATE 17, 30: PRINT " A   B   C   D   E   "
```

```
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER  ", R$
```

```
x2$ = TIME$
```

```
IF R$ = "D" THEN score = score + 1
```

```
GOSUB 1000
```

```
CLS
```

```
REM problem 3
```

```
NS = NS + 1
```

```
x1$ = TIME$
```

```
GOSUB 150
```

```
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
```

```
LOCATE 10, 30: PRINT " i)All books are calculator."
```

```
LOCATE 11, 30: PRINT " ii)No calculator is scientific."
```

```
LOCATE 13, 30: PRINT " CONCLUSIONS"
```

```
LOCATE 14, 30: PRINT " a) Some books are scientific "
```



## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 15, 30: PRINT " b) No book is scientific"
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 4
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)Some dogs are black."
LOCATE 11, 30: PRINT " ii)Some black are horses."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a) Some horses are dogs "
LOCATE 15, 30: PRINT " b) No horse is black"
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 5
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)Some cats are tiger."
LOCATE 11, 30: PRINT " ii)All tigers are lions."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a) Some cats are lions. "
LOCATE 15, 30: PRINT " b) Some lions are cats."
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 6
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)All biscuits are chocolates."
LOCATE 11, 30: PRINT " ii)No chocolate is an ice-cream."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a)No biscuit is a chocolate"
LOCATE 15, 30: PRINT " b) All chocolates are biscuits."
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
```

## Analysis of Computer programming tasks and Aptitudes

```
GOSUB 1000
CLS
```

```
REM problem 7
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)All elephants are birds."
LOCATE 11, 30: PRINT " ii)Some birds are cows. "
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a) Some cows are birds. "
LOCATE 15, 30: PRINT " b) Some elephants are cows. "; ""
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 8
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)All papers are pencils."
LOCATE 11, 30: PRINT " ii)All pencils are erasers."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a) Some erasers are papers."
LOCATE 15, 30: PRINT " b) Some pencils are not papers."
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 9
NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)Some kites are horses."
LOCATE 11, 30: PRINT " ii)All horses are dogs."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a)All dogs are horses."
LOCATE 15, 30: PRINT " b) Some dogs are horses."
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 10
```

## Analysis of Computer programming tasks and Aptitudes

```

NS = NS + 1
x1$ = TIME$
GOSUB 150
COLOR 14, 9, 4: LOCATE 9, 30: PRINT " STATEMENTS"
LOCATE 10, 30: PRINT " i)Some trees are flowers."
LOCATE 11, 30: PRINT " ii)Some ships are trees."
LOCATE 13, 30: PRINT " CONCLUSIONS"
LOCATE 14, 30: PRINT " a)Some flowers are ships."
LOCATE 15, 30: PRINT " b)Some ships are trees."
LOCATE 17, 30: PRINT " A B C D E "
LOCATE 19, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM COLOR 2, 8, 11
SCREEN 0
PRINT "Directions:"
PRINT " "
REM COLOR 2, 8, 11
PRINT "Each of the items below consists of a question and
statements numbered (a) and (b). You have to decide whether the data
provided in the statements are sufficient to answer the
question. Read both the statements and give answer as follows:"
PRINT
PRINT "(A) if statement 'a' alone is sufficient but statement 'b'
is not sufficient."
PRINT "(B) if statement 'b' alone is sufficient but statement 'a'
is not sufficient ."
PRINT "(C)if either statement 'a' or 'b' is sufficient."
PRINT "(D)if given data is not sufficient "
PRINT "(E)if statements 'a' and 'b' together are necessary."
PRINT
INPUT C

CLS
LOCATE 3, 20: PRINT "FOR EXAMPLE "
COLOR 14, 9, 4: LOCATE 5, 20: PRINT " How many students in a school
eats in the canteen?"
LOCATE 7, 20: PRINT " a)Half the girls in the school eat in the
canteen."
LOCATE 8, 20: PRINT "b)One-third of the boys in the school eat in
the canteen."
LOCATE 12, 20: PRINT " A B C D E "
LOCATE 15, 20: INPUT "TYPE YOUR ANSWER ", R$
PRINT "Here both the statements are not sufficient to answer the
question. Hence the correct answer is (E)."
```

INPUT " DO YOU UNDERSTAND (IF YES, PRESS KEY 1) ", A

IF A <> 1 GOTO 10

CLS

```

PRINT #3, "*****INFERENCE TEST 2*****"
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"
```

## Analysis of Computer programming tasks and Aptitudes

```
180 NS = 0
REM problem 11
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 20: PRINT " Does light travel faster than sound?"
LOCATE 12, 20: PRINT " a) Light can travel in vacuum"
LOCATE 13, 20: PRINT " b) Sound cannot travel in vacuum."
LOCATE 15, 20: PRINT " A B C D E "
LOCATE 17, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 12
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 5: PRINT " How many restaurants are there in Kolkata?"
LOCATE 11, 20: PRINT " a)You take three meals a day and every meal
is in a restaurant"
PRINT " you have not visited before."
LOCATE 14, 20: PRINT " b)It takes you 3 years to cover all the
restaurants in Delhi."
LOCATE 17, 20: PRINT " A B C D E "
LOCATE 19, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

REM problem 13
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 20: PRINT " There are 30 memebrs in a club.How many
members are female lawyers?"
LOCATE 13, 20: PRINT "a)10 members are ladies."
LOCATE 14, 20: PRINT "b)20 members are lawyers"
LOCATE 16, 20: PRINT " A B C D E "
LOCATE 18, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

REM problem 14
NS = NS + 1
x1$ = TIME$
```

## Analysis of Computer programming tasks and Aptitudes

```
GOSUB 160
LOCATE 10, 20: PRINT " Who is Lata's sister?"
LOCATE 13, 20: PRINT " a)Abinash is the brother of Lata."
LOCATE 14, 21: PRINT "b)Lata's father is the husband of Abinash's
mother."
LOCATE 16, 20: PRINT " A   B   C   D   E   "
LOCATE 18, 21: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

REM problem 15
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 5: PRINT " In which year Rupa passed 8th standard of
examination?"
LOCATE 12, 20: PRINT " a)At present Rupa is reading at thee
standards higher than her"
PRINT "   younger brother."
LOCATE 14, 20: PRINT " b)Her younger brother was admitted in class I
in the year 1997."
LOCATE 17, 20: PRINT " A   B   C   D   E   "
LOCATE 19, 20: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000
CLS

REM problem 16
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 10: PRINT " In which day of the week did Prabir meet with
an accident?"
LOCATE 12, 10: PRINT " (a)Prabir was relieved from hospital on
Saturday,three days"
PRINT "           after he met with the accident."
LOCATE 14, 10: PRINT "(b)Prabir was travelling on a scooter on
Monday."
LOCATE 16, 10: PRINT " A   B   C   D   E   "
LOCATE 18, 10: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "E" THEN score = score + 1
GOSUB 1000

CLS
REM problem 17
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 20: PRINT " In which year was Tarun born?"
LOCATE 12, 12: PRINT "   I)Tarun is six years older than Rabin."
```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 13, 12: PRINT " II)Rabin's brother was born in 1982."
LOCATE 14, 12: PRINT " III)Tarun's brother is two years younger
than"
PRINT " Rabin's brother who was eight years younger
than him."
LOCATE 17, 16: PRINT " A) II and III only "
PRINT " B) I and III only "
PRINT " C)I, II and III "
PRINT " D) I and II only "
PRINT " E)None of these. "
LOCATE 23, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 18
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 10, 2: PRINT "WHO AMONG P,Q,R,S AND T IS IN THE MIDDLE WHILE
STANDING IN A ROW?"
LOCATE 12, 20: PRINT " I)Q is to the right of T."
LOCATE 13, 20: PRINT " II)S is between P and T."
LOCATE 14, 20: PRINT " III)Q is between T and R."
LOCATE 16, 20: PRINT " A) I and II only "
PRINT " B) I and III only "
PRINT " C) II and III only "
PRINT " D) I, only "
PRINT " E)None of these. "
LOCATE 22, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 19
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 8, 25: PRINT "Who among Girish,Prasant,Kamlesh and Jayesh is
the youngest?"
LOCATE 11, 19: PRINT " A)Kamlesh is younger than Jayesh but older"
PRINT " than Girish and Prasant"
LOCATE 13, 20: PRINT "B)Jayesh is the eldest."
LOCATE 14, 19: PRINT " C)Girish is older than Prasant."
LOCATE 16, 19: PRINT " A) I and II only "
PRINT " B)II and III only "
PRINT " C) I and III only "
PRINT " D) I, II and III "
PRINT " E) None of these. "
LOCATE 22, 30: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
```

## Analysis of Computer programming tasks and Aptitudes

```

GOSUB 1000
CLS

REM problem 20
NS = NS + 1
x1$ = TIME$
GOSUB 160
LOCATE 9, 20: PRINT "How many sons does D have?"
LOCATE 12, 20: PRINT " I) B and F are brothers of E."
LOCATE 13, 21: PRINT "II) C is the sister of A and F."
LOCATE 14, 21: PRINT "III) C and E are daughters of D."
LOCATE 16, 21: PRINT " A) I, II and III  "
PRINT "                B)II and III only  "
PRINT "                C) I and II only  "
PRINT "                D) I, II and III together are not
sufficient.  "
PRINT "                E) None of these.  "
LOCATE 23, 30: INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

SCREEN 0, 1
COLOR 14, 9, 4
LOCATE 10, 35: PRINT "THANK YOU"
CLOSE #3
INPUT "location of index file", fiori$
RUN fiori$

CLS
1000 REM "CALCULATING TIME"
BEEP
sec1$ = RIGHT$(x1$, 2)
sec1 = VAL(sec1$)
sec2$ = RIGHT$(x2$, 2)
sec2 = VAL(sec2$)
IF sec2 < sec1 THEN td = (60 - sec1) + sec2
IF sec2 > sec1 THEN td = sec2 - sec1
PRINT #3, NS, R$, x1$, x2$, td, score
RETURN

150 REM REPEATED TEXT
PRINT "GIVE ANSWER          "
PRINT "                (A) IF ONLY CONCLUSION (a)
FOLLOWS."
PRINT "                (B) IF ONLY CONCLUSION (b) FOLLOWS"
PRINT "                (C) IF EITHER (a) OR (b) FOLLOWS"
PRINT "                (D) IF NEITHER (a) NOR (b) FOLLOWS"
PRINT "                (E) IF BOTH (a) AND (b) FOLLOWS."
PRINT "-----"
PRINT "-"
RETURN

```

## Analysis of Computer programming tasks and Aptitudes

```
160 REM REPEATED TEXT 2
PRINT "GIVE ANSWER      "
PRINT "(A) if statement 'a' alone is sufficient but  statement 'b'
is not sufficient."
PRINT "(B) if statement 'b' alone is sufficient but  statement 'a'
is not sufficient ."
PRINT "(C)if either statement 'a' or 'b' is sufficient."
PRINT "(D)if given data is not sufficient "
PRINT "(E)if statements 'a' and 'b' together are necessary."
PRINT "-----"
-----"
RETURN
```



## Analysis of Computer programming tasks and Aptitudes

### MEMORY FOR COMMANDS

```
DIM NAME1$(10), MEAN1$(10), NAME2$(10), MEAN2$(10), NAME3$(10),
MEAN3$(10)
CLS
REM This is a test for measurement of MEMORY

LOCATE 1, 4: PRINT DATE$
LOCATE 1, 60: PRINT TIME$
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3

LOCATE 12, 5: PRINT "MEMORY TEST"

GOSUB 20: REM DIRECTION
GOSUB 30: REM READING NAME AND MEANING
CLS
LOCATE 10, 15: INPUT "ARE YOU READY ? ", R
CLS
LOCATE 15, 15: PRINT "MEMORY TEST STARTING"
SLEEP 2
GOSUB 40: REM DATA COLLECTION
LOCATE 5, 20: PRINT "total score =", SCORE1
SLEEP 2
SCORE = 0

GOSUB 45
GOSUB 50: REM LEARNING PAIRED ASSOCIATED COMMANDS
CLS
LOCATE 10, 15: INPUT "ARE TOU READY ?", S
CLS
LOCATE 15, 15: PRINT "MEMORY TEST STARTING"
SLEEP 2
GOSUB 60
GOSUB 70
GOSUB 80: REM READING PAIRED ASSOCIATED COMMANDS AND THEIR MEANING
CLS
LOCATE 10, 15: INPUT "ARE YOU READY ?", T
CLS
LOCATE 15, 15: PRINT "MEMORY TEST STARTING"
SLEEP 2
GOSUB 90
PRINT #3, "TOTAL SCORE = ", SCORE1 + SCORE2 + SCORE3
LOCATE 20, 30: PRINT "THANK YOU"
SLEEP 2
CLOSE #3
INPUT "LOCATION OF INDEX FILE", FIORI$
RUN FIORI$

REM *****END OF MAIN
PROGRAM*****

10 REM INTRODUCTION
COLOR 2, 8, 11
PRINT #3, "PROBLEM NO"; " STIMULUS:", " RESPONSE:", " SCORE"
```

## Analysis of Computer programming tasks and Aptitudes

```
20 REM DIRECTION
CLS
LOCATE 8, 30: PRINT "A PAIRED ASSOCIATED LIST OF COMMANDS AND THEIR
MEANINGS WILL BE PRESENTED TO YOU.PLEASE REMEMBER THEM. IN MEMORY
TEST, ONLY THE MEANING WILL BE PRESENTED AND YOU ARE REQUESTED TO
TYPE THE ASSOCIATED COMMAND ON THE TERMINAL"
LOCATE 20, 30: INPUT "Have you finished ? ", h
COLOR 7, 8, 11
LOCATE 20, 10: PRINT " PLEASE USE UPPER CASE ALPHABET IN RESPONDING"
LOCATE 22, 10: INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", v
CLS
SCREEN 0
PLAY "abcabc"
RETURN
```

```
30 REM READING NAME AND MEANING
CLS
COLOR 7, 8, 11
FOR I = 1 TO 10
READ NAME1$(I), MEAN1$(I)
LOCATE 10, 4: PRINT I
COLOR 14, 9, 4: LOCATE 10, 5: PRINT "COMMAND": COLOR 2, 8, 11:
LOCATE 10, 15: PRINT NAME1$(I)
COLOR 14, 9, 4: LOCATE 12, 5: PRINT "MEANING": COLOR 2, 8, 11:
LOCATE 12, 15: PRINT MEAN1$(I)
SLEEP 5
REM LOCATE 20, 2: INPUT " OK ? ", OK
CLS
NEXT I

DATA INPUT , ENTER SOME DATA IN TERMINAL
DATA CONTINUE , ASKING THE PROGRAM TO CARRY ON EXECUTION
DATA FORMAT , SPECIFYING THE ARRANGEMENT OF OUTPUT DATA
DATA DIMENSIONS, SPECIFYING MEMORY LOCATIONS
DATA GOSUB , GO TO SUBROUTINE
DATA CHARACTER, STRING EXPRESSION
DATA CASE, SELECTING A GROUP OF STATEMENTS OUT OF
ALTERNATIVES
DATA INTEGER, WHOLE NUMBER
DATA GOTO, SPECIFYING LOOP
DATA COMMON, TRANSMITTING VARIABLES BETWEEN MAIN PROGRAM AND
SUBROUTINE
RETURN
```

```
40 REM COLLECTION OF DATA ( COMMAND AND MEANING)
PRINT #3, "-----PAIRED ASSOCIATED COMMAND DATA ( COMMAND AND
MEANING)----- "
CLS
FOR I = 1 TO 10
LOCATE 12, 5: COLOR 2, 8, 11: PRINT MEAN1$(I)
LOCATE 15, 5: COLOR 14, 9: INPUT "TYPE ASSOCIATED COMMAND ", COM$
IF COM$ = NAME1$(I) THEN SCORE1 = SCORE1 + 1
```

## Analysis of Computer programming tasks and Aptitudes

```

PRINT #3, I; " ** "; MEAN1$(I); " ** "; COM$; "***"; SCORE1
CLS
NEXT I
PRINT #3, "PAIRED ASSOCIATED (COMMAND AND MEANING) SCORE=", SCORE1

RETURN

45 REM DIRECTION
CLS
LOCATE 8, 30: PRINT " A PAIRED ASSOCIATED LIST OF COMMANDS WILL BE
PRESENTED TO YOU.PLEASE REMEMBER THEM. IN MEMORY TEST, ONLY ONE
COMMAND WILL BE PRESENTED AND YOU ARE REQUESTED TO TYPE THE
ASSOCIATED COMMAND ON THE TERMINAL"
LOCATE 20, 5: INPUT "DO YOU UNDERSTAND ? ", O
RETURN

50 REM LEARNING PAIRED ASSOCIATED COMMANDS
CLS
COLOR 7, 8, 11
FOR I = 1 TO 10
READ NAME2$(I), MEAN2$(I)
LOCATE 10, 4: PRINT I
COLOR 14, 9, 4: LOCATE 10, 5: PRINT "COMMAND1 ": COLOR 2, 8, 11:
LOCATE 10, 15: PRINT NAME2$(I)
COLOR 14, 9, 4: LOCATE 12, 5: PRINT "COMMAND2 ": COLOR 2, 8, 11:
LOCATE 12, 15: PRINT MEAN2$(I)
SLEEP 3
CLS
NEXT I

DATA OPEN , LOAD
DATA READ , SCANf
DATA CONTINUE , ENDO
DATA INCLUDE, IMPORT
DATA QUIT , CLS
DATA EXIT , fCLOSE
DATA EDIT , Vi
DATA FOR , DO
DATA DISPLAY , fPRINTF
DATA CAT , DIR
RETURN

60 REM COLLECTION OF DATA ( TWO COMMANDS )
PRINT #3, "-----PAIRED ASSOCIATED COMMAND DATA ( TWO COMMANDS )-----
"
CLS
FOR I = 1 TO 10
CLS
LOCATE 10, 5: COLOR 2, 8, 11: PRINT MEAN2$(I)
LOCATE 12, 5: COLOR 14, 9: INPUT "TYPE ASSOCIATED COMMAND ", COM$
IF COM$ = NAME2$(I) THEN SCORE2 = SCORE2 + 1
REM COLOR 7, 8, 11
PRINT #3, I; "***"; , "***"; MEAN2$(I); "***"; COM$; "***"; SCORE2
NEXT I

```

## Analysis of Computer programming tasks and Aptitudes

```

PRINT COM$, "TWO COMMANDS SCORE = ", SCORE2
SLEEP
SCORE = 0
CLS
PRINT #3, "PAIRED ASSOCIATED (TWO COMMANDS) SCORE=", SCORE2
RETURN

70 REM DIRECTION
LOCATE 8, 30: PRINT " A PAIRED ASSOCIATED LIST OF COMMANDS AND THEIR
MEANINGS WILL BE PRESENTED TO YOU.PLEASE REMEMBER THEM. IN MEMORY
TEST, ONLY THE MEANING WILL BE PRESENTED AND YOU ARE REQUESTED TO
TYPE THE ASSOCIATED COMMAND ON THE TERMINAL"
LOCATE 20, 5: INPUT "DO YOU UNDERSTAND ? ", O
RETURN

80 REM READING PAIRED ASSOCIATED COMMANDS AND THEIR MEANINGS
CLS
PRINT
COLOR 7, 8, 11
FOR I = 1 TO 10
READ NAME3$(I), MEAN3$(I)
LOCATE 10, 5: COLOR 14, 9, 4: PRINT "COMMAND": COLOR 2, 8, 11:
LOCATE 10, 15: PRINT NAME3$(I)
LOCATE 12, 5: COLOR 14, 9, 4: PRINT "STATEMENT": COLOR 2, 8, 11:
LOCATE 12, 15: PRINT MEAN3$(I)
PRINT
PRINT
SLEEP 4
CLS
NEXT I

DATA ^B , REFORM PARAGRAPH
DATA ^E , MOVE UP
DATA ^T , DELETE WORD
DATA ^D , MOVE TO RIGHT
DATA ^L , FIND AND REPLACE
DATA ^QL, CHANGE SPELLING
DATA ^KM, ADD NUMBER
DATA ^KE, CHANGE NAME
DATA ^QO, TYPE AND CHECK
DATA ^OE, INSERT HYPHEN
RETURN

90 REM COLLECTION OF DATA ( CONTROL AND MEANING )
CLS
FOR I = 1 TO 10
REM PRINT i, MEAN3$(i): INPUT "TYPE ASSOCIATED COMMAND ", COM$
CLS
LOCATE 10, 10: COLOR 2, 8, 11: PRINT MEAN3$(I)
LOCATE 12, 5: COLOR 14, 9: INPUT "TYPE ASSOCIATED COMMAND ", COM$
IF COM$ = NAME3$(I) THEN SCORE3 = SCORE3 + 1
COLOR 7, 8, 11
PRINT #3, I; "***"; MEAN3$(I); "***"; COM$; "***"; SCORE3
NEXT I
LOCATE 15, 20: PRINT "total score =", SCORE3

```

## Analysis of Computer programming tasks and Aptitudes

RETURN

### VOCABULARY TEST

```
CLS
REM This is a test for measurement of vocabulary
LOCATE 1, 4: PRINT DATE$
LOCATE 1, 60: PRINT TIME$
file$ = "c:\windows\desktop\apti.out"
OPEN file$ FOR APPEND AS #3
COLOR 2, 8, 11
PRINT #3, " ***** VOCABUL TEST *****"
PRINT #3, "PROBLEM NO"; " RESPONSE:"; " INITIAL TIME:"; " FINAL
TIME:"; " TIME DIFFERENCE:"; " SCORE"

CLS
LOCATE 10, 30: PRINT " VOCABULARY TEST  "
COLOR 7, 8, 11
LOCATE 20, 10: PRINT " PLEASE USE UPPER CASE ALPHABETS IN
RESPONDING"
LOCATE 22, 10: INPUT " HAVE YOU LOCKED CAPITAL LETTERS ? ", V
SLEEP 2
CLS
SCREEN 0
PLAY "abcabc"
10 LOCATE 2, 30: PRINT "SECTION A "
LOCATE 4, 5: PRINT "Directions:"
LOCATE 6, 5: PRINT "THERE ARE 15 ITEMS IN THIS SECTION. EACH ITEM
CONSISTS OF A WORD FOLLOWED BY 4 WORDS,i.e, a, b, c,and d. PLEASE
SELECT THE WORD WHICH IS MOST NEARLY SIMILAR IN MEANING TO THE
ORIGINAL WORD"
LOCATE 11, 4: PRINT " For example"
LOCATE 13, 4: PRINT " HIGH:"
LOCATE 15, 4: PRINT " a)TALL          b)SHORT          c)FAT
d)THIN          "
PRINT " "
PRINT " Here the correct answer is 'A', so you just press the key
'A' "
INPUT "Do you understand the direction ? ( yes = 1 ) ", a
IF a <> 1 GOTO 10

REM TESTING STARTS NOW
CLS
NS = 0
NS = NS + 1
x1$ = TIME$
REM Problem 1
LOCATE 10, 20: PRINT " JOURNEY : "
LOCATE 12, 20: PRINT "a)RUN  b)RELAX  c)JUMP  d)TOUR "
LOCATE 20, 20: INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
```

## Analysis of Computer programming tasks and Aptitudes

CLS

```
REM problem 2
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT " RISKY: "
LOCATE 12, 20: PRINT " a)STUPID b)VIOLENT c)DANGEROUS d)EXCITING"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 3
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "LIKELY: "
LOCATE 12, 20: PRINT " a)BRIGHT b) PROBABLE c)BEST-SUITED
d)USUAL "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 4
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT " CONVICTED : "
LOCATE 12, 20: PRINT " a) JAILED b)ASHAMED c)DEPENDENT d)DAMAGED
"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 5
NS = NS + 1
x1$ = TIME$
LOCATE 10, 2: PRINT " DATA:"
LOCATE 12, 2: PRINT " a) SCIENTIFIC THEORY b)TECHNICAL KNOWLEDGE
c)INFORMATION d)COMPUTER LANGUAGE "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 6
NS = NS + 1
```

## Analysis of Computer programming tasks and Aptitudes

```
x1$ = TIME$
LOCATE 10, 20: PRINT "FASCINATING:"
LOCATE 12, 20: PRINT " a) CHARMING b)BIG c)DEMOCRATIC d)POWERFUL"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 7
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "TRANSPARENT:"
LOCATE 12, 20: PRINT " a)WORDY b)CLEAR c)INVOLVED d)WITTY"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 8
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "GRAVITY:"
LOCATE 12, 20: PRINT " a) HUMBLENESS b)LENGTH c)JOY d)SERIOUSNESS
"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 9
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT " CONSEQUENCE : "
LOCATE 12, 20: PRINT " a)SERIAL b)RESULT c)CAUSE d)FORCEFUL "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 10
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT " VERBATIM: "
LOCATE 12, 20: PRINT " a)DISCUSSION b)UNEDITED c)INFORMATION
d)OPPURTUNITY "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
```

## Analysis of Computer programming tasks and Aptitudes

```
GOSUB 1000
CLS

REM problem 11
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "  RECALCITRANT:  "
LOCATE 12, 20: PRINT " a) SIMULTANEOUS  b) PERMANENT  c) UNOBTAINABLE
d) STUBBORN"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 12
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "  RETICENT:  "
LOCATE 12, 20: PRINT " a) TENDER  b) RESERVED  c) GLORIFYING
d) GLOOMY"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 13
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "  OBSEQUIOUS:"
LOCATE 12, 20: PRINT " a) SERVILE  b) POOR  c) CHEERFUL  d) SULLEN"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

REM problem 14
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "HALLOWED:"
LOCATE 12, 20: PRINT " a) HISTORICAL  b) ANCIENT  c) SACRED  d) DECAYED
"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

REM problem 15
NS = NS + 1
x1$ = TIME$
```



## Analysis of Computer programming tasks and Aptitudes

```

LOCATE 10, 20: PRINT "INOPPORTUNE: "
LOCATE 12, 20: PRINT " a) PREMEDIATE b) FORTUITOUS c) IMPROMPTU
d) UNTIMELY "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

20 COLOR 2, 8, 11
LOCATE 1, 35: PRINT "SECTION B "
LOCATE 4, 5: PRINT " DIRECTIONS : "
LOCATE 6, 5: PRINT " THERE ARE 15 ITEMS IN THIS SECTION. EACH ITEM
CONSISTS OF A WORD FOLLOWED BY 4 WORDS THAT IS a b c d. SELECT THE
WORD WHICH IS MOST NEARLY OPPOSITE IN MEANING TO THE
ORIGINAL WORD"
LOCATE 11, 5: PRINT " FOR EXAMPLE "
LOCATE 13, 5: PRINT " DAY:"
LOCATE 15, 5: PRINT " a) YEAR b) MONTH c) NIGHT d) HOUR"
LOCATE 17, 5: PRINT " HERE THE CORRECT ANSWER IS 'c', SO YOU HAVE TO
PRESS THE KEY 'c'"
LOCATE 20, 5: INPUT " DO YOU UNDERSTAND THE DIRECTIONS ? (IF YES,
PRESS 1)", a
IF a <> 1 GOTO 20

CLS
REM problem 16
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "BIRTH : "
LOCATE 12, 20: PRINT " a) ANNIVERSARY b) FESTIVAL c) DEATH d) LIFE
"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "C" THEN score = score + 1
GOSUB 1000
CLS

REM problem 17
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "GENUINE:"
LOCATE 12, 20: PRINT " a) FOOLISH b) UNREAL c) UGLY d) SMOOTH "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 18
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "PRIVATE:"

```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 12, 20: PRINT " a) PERSONAL b)ROUGH c)PUBLIC d)AVERAGE"  
LOCATE 20, 20  
INPUT "TYPE YOUR ANSWER ", R$  
x2$ = TIME$  
IF R$ = "C" THEN score = score + 1  
GOSUB 1000  
CLS
```

```
REM problem 19  
NS = NS + 1  
x1$ = TIME$  
LOCATE 10, 20: PRINT "NATIVE:"  
LOCATE 12, 20: PRINT " a)LOVING b)STRONG c)FOREIGNER d)MILD"  
LOCATE 20, 20  
INPUT "TYPE YOUR ANSWER ", R$  
x2$ = TIME$  
IF R$ = "C" THEN score = score + 1  
GOSUB 1000  
CLS
```

```
REM problem 20  
NS = NS + 1  
x1$ = TIME$  
LOCATE 10, 20: PRINT "CONFIRM: "  
LOCATE 12, 20: PRINT " a) AVOID b)DENY c)REFUSE d)REJECT"  
LOCATE 20, 20  
INPUT "TYPE YOUR ANSWER ", R$  
x2$ = TIME$  
IF R$ = "B" THEN score = score + 1  
GOSUB 1000  
CLS
```

```
REM problem 21  
NS = NS + 1  
x1$ = TIME$  
LOCATE 10, 20: PRINT "FRAIL: "  
LOCATE 12, 20: PRINT " a)FURIOUS b)ROUGH c)SIMPLE d)STOUT"  
LOCATE 20, 20  
INPUT "TYPE YOUR ANSWER ", R$  
x2$ = TIME$  
IF R$ = "D" THEN score = score + 1  
GOSUB 1000  
CLS
```

```
REM problem 22  
NS = NS + 1  
x1$ = TIME$  
LOCATE 10, 20: PRINT "FLEXIBLE:"  
LOCATE 12, 20: PRINT " a) WEAK b)HOSTILE c)UNCOMPROMISING  
d)NEUTRAL "  
LOCATE 20, 20  
INPUT "TYPE YOUR ANSWER ", R$  
x2$ = TIME$  
IF R$ = "C" THEN score = score + 1  
GOSUB 1000  
CLS
```

## Analysis of Computer programming tasks and Aptitudes

```
REM problem 23
NS = NS + 1
x1$ = TIME$
LOCATE 10, 2: PRINT "DEPRESSING:"
LOCATE 12, 2: PRINT " a) MAKING HAPPIER  b) HORRIFYING  c) BEWILDERING
d) EQUIVALENT "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 24
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "RELUCTANT:"
LOCATE 12, 20: PRINT " a) AFRAID  b) EAGER  c) HELPFUL  d) LOGICAL "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 25
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "RECOLLECT:"
LOCATE 12, 20: PRINT " a) MEMORY  b) DISTRIBUTE  c) REARRANGE
d) FORGET "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 26
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "NEPOTISM:"
LOCATE 12, 20: PRINT " a) INDIFFERENCE  b) IMPARTIALITY  c) HATRED
d) CONDEMNATION "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER  ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS
```

```
REM problem 27
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "MUNDANE:"
```

## Analysis of Computer programming tasks and Aptitudes

```
LOCATE 12, 20: PRINT " a) SUPERB b) HEAVENLY c) FRIENDLY
d) DECENCY "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "B" THEN score = score + 1
GOSUB 1000
CLS

REM problem 28
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "PARSIMONIOUS:"
LOCATE 12, 20: PRINT " a) LAVISH b) UNEDUCATED c) DIVERGENT
d) STUPID "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS

REM problem 29
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "LETHARGY:"
LOCATE 12, 20: PRINT " a) ACIDITY b) RIGIDITY c) SOLITARY
d) ALERTNESS "
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "D" THEN score = score + 1
GOSUB 1000
CLS

REM problem 30
NS = NS + 1
x1$ = TIME$
LOCATE 10, 20: PRINT "ENIGMATIC:"
LOCATE 12, 20: PRINT " a) CLEAR b) INEXPENSIVE c) OBSTINATE
d) CURIOUS"
LOCATE 20, 20
INPUT "TYPE YOUR ANSWER ", R$
x2$ = TIME$
IF R$ = "A" THEN score = score + 1
GOSUB 1000
CLS
LOCATE 5, 20: PRINT "total score =", score
LOCATE 20, 30: PRINT "THANK YOU"
GOTO 2000

CLS
1000 REM "CALCULATING TIME"
BEEP
sec1$ = RIGHT$(x1$, 2)
```

## Analysis of Computer programming tasks and Aptitudes

```
sec1 = VAL(sec1$)

sec2$ = RIGHT$(x2$, 2)
sec2 = VAL(sec2$)
IF sec2 < sec1 THEN td = (60 - sec1) + sec2
IF sec2 > sec1 THEN td = sec2 - sec1
PRINT NS, R$, x1$, x2$, td, score
RETURN

2000 CLOSE #3
INPUT "LOCATION OF INDEX FILE", FIORI$
RUN FIORI$
```

**NUMBER SERIES TEST**

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>
FILE *fp;
struct student{
char d;
char s_code[6];
char s_name[20];
char s_age[2];
char s_edu[15];
char s_ocu[15];
char inst_n[30];
char inst_a[40];
char s_add[40];
char s_exp[15];
char s_co[15];
char s_cd[20];
}S;

main(){
int i,j;
clrscr();
textbackground(BLACK);
for(i=10;i<16;i++){
for(j=16;j<68;j++){
gotoxy(j,i);
printf("%c",220);
delay(5);
}
}
for(i=11;i<15;i++){
for(j=18;j<65;j++){
gotoxy(j,i);
printf("%c",255);
delay(5);
}
}
for(i=11;i<15;i++){
for(j=18;j<53;j++){
gotoxy(20,13);
printf(" WELCOME TO INDIAN STATISTICAL INSTITUTE ");
delay(20);
}
}
clrscr();
add();
clrscr();
gotoxy(30,4);
printf("NUMBER SERIES");
gotoxy(31,6);
printf("DIRECTIONS");

```

## Analysis of Computer programming tasks and Aptitudes

```
gotoxy(10,8);
printf("This test consists of 20 numerical problems.Next to each ");
gotoxy(10,10);
printf("problem there are four answers.These are marked by A,B,C,");
gotoxy(10,12);
printf("and D.Only one of these suggested answer is right.You are");
gotoxy(10,14);
printf("to pick out the one correct answer for each problem.Then ");
gotoxy(10,16);
printf("write the answer mark from keybord.And press enter for
next");
gotoxy(10,18);
printf("question.");
gotoxy(10,23);
printf("Press any key to continue....");
getch();
clrscr();
test();
return 0;
}
```

```
add() {
    FILE *fp;
    long int l;
    char ans='Y';
    fp=fopen("Student1.txt","a");
    fseek(fp,0L,2);
    l=ftell(fp);
    l=l/(long int)sizeof(struct student);
    sprintf(S.s_code,"%04ld",++l);
    gotoxy(10,2);
    printf("User No:%s",S.s_code);
    gotoxy(10,4);
printf("Enter user name:");
gets(S.s_name);
gotoxy(10,6);
printf("Enter user age:");
gets(S.s_age);
gotoxy(10,8);
printf("Enter user Education:");
gets(S.s_edu);
gotoxy(10,10);
printf("Enter user ocupation:");
gets(S.s_ocu);
gotoxy(10,12);
printf("Enter user institute name:");
gets(S.inst_n);
gotoxy(10,14);
printf("Enter user institute address:");
gets(S.inst_a);
gotoxy(10,16);
printf("Enter user Address:");
gets(S.s_add);
gotoxy(10,18);
printf("Enter user experience:");
gets(S.s_exp);
gotoxy(10,20);
```

## Analysis of Computer programming tasks and Aptitudes

```

printf("Enter the course name which you done:");
gets(S.s_co);
gotoxy(10,22);
printf("Enter name of certificate or diploma:");
gets(S.s_cd);
clrscr();
gotoxy(10,12);
printf("Press any key for continue(Press N for exit)..");
ans=toupper(getche());
if(ans!='N'){
fwrite(&S,sizeof(S),1,fp);
fclose(fp);
}
return 0;
}

/*TESTING PARTS*/
test(){
char a;
clrscr();
gotoxy(8,2);
printf("DEMO TEST :");
gotoxy(8,4);
printf("Which one of the following numbers will complete the series
?");
gotoxy(8,6);
printf("2, 4, 6, ____,10");
gotoxy(8,8);
printf("A.12 B.8 C.14 D.16\n");
gotoxy(8,10);
printf("In example 8 is the correct answer,So you have just write
B.");
gotoxy(8,12);
printf("Now you try the example bellow....");
gotoxy(8,14);
printf("TEST :");
gotoxy(8,16);
printf("7, 9, 11, ____,15");
gotoxy(8,18);
printf("A.17 B.13 C.5 D.10\n");
a=toupper(getche());
if(a=='B')
exam();
else {
clrscr();
printf("Sorry You Not Passed In Test...Please Try Later ...!");
}
return 0;
}

/* EXAM PARTS */
exam()
{
int x,y,z,i,j,r,ans,result=0;
float af,f,f1;
char c,a,ch[4]={'A','B','C','D'};
randomize();

```



## Analysis of Computer programming tasks and Aptitudes

```
    /* Question No.1 */
clrscr();
gotoxy(10,10);
printf("Q1. ");
fflush(stdin);
i=random(6);
i=i+1;
for(j=0;j<7;j++){
    if(j==6){
        printf("__ ");
        ans=i;
    }
    else {
printf("%d, ",i);
i=i+2;
    }
}
gotoxy(10,14);
fflush(stdin);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=random(20);
if(i<14)i=i+11;
if((i==ans)|| (i==0))
i++;
printf("%c)%d    ",ch[j],i);
}
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

    /* Question No. 2 */
clrscr();
x=3;
gotoxy(10,10);
fflush(stdin);
printf("Q2. ");
i=random(5);
i=i+1;
for(j=0;j<6;j++){
    if(j==5){
        printf("__ ");
        ans=i;
    }
    else {
printf("%d, ",i);
i=i+x;
x=x*2;
    }
}
gotoxy(10,14);
```

```

fflush(stdin);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=random(100);
if(i==ans)
i++;
/*  if((i==0)|| (i<90))
i=ans+i;*/
printf("%c)%d    ",ch[j],i);
}
printf("\n");
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

        /*Question No. 3*/
clrscr();
gotoxy(10,10);
printf("Q3. ");
fflush(stdin);
i=random(4);
i++;
for(j=0;j<5;j++){
if(j==4){
printf("__ ");
ans=i*i;
}
else{
printf("1/%d, ",i*i);
i++;
}
}
gotoxy(10,14);
fflush(stdin);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)1/%d    ",ch[j++],ans);
break;
}
i=random(60);
if(i==ans);
i++;
printf("%c)1/%d    ",ch[j],i);
}
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);

```

```
getch();
```

```

    /* Question No. 4 */
    clrscr();
    x=9;
    gotoxy(10,10);
    printf("Q4. ");
    fflush(stdin);
    i=random(6);
    i++;
    for(j=0;j<4;j++){
        if(j==3){
            printf("__ ");
            ans=i;
        }
        else{
            printf("%d, ",i);
            i=(i*2)+2;
        }
    }
    gotoxy(10,14);
    fflush(stdin);
    r=random(3);
    for(j=0;j<4;j++){
        while(j==r){
            c=ch[j];
            printf("%c)%d    ",ch[j++],ans);
            break;
        }
        i=random(ans+10);
        if(i==ans);
        i++;
        printf("%c)%d    ",ch[j],i);
    }
    a=toupper(getche());
    if(c==a)
        result++;
    printf("\nResult : %d",result);
    getch();

```

```

    /*Question No. 5 */
    clrscr();
    gotoxy(10,10);
    printf("Q5. ");
    fflush(stdin);
    i=random(4);
    i++;
    r=random(3);
    for(j=0;j<6;j++){
        if(j==5){
            printf("__ ");
            ans=i;
        }
        else{
            printf("%d, ",i);

```

```

i=i+3;
}
}
gotoxy(10,14);
fflush(stdin);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=random(ans+10);
if(i==ans);
i++;
printf("%c)%d    ",ch[j],i);
}
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 6 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q6. Which one number is wrong in the following series
?");
gotoxy(12,13);
i=random(5);
i++;
x=i;
r=random(5);
for(j=0;j<5;j++){
if(j==r){
printf("%d, ",r=i+3);
ans=r;
i=(i*2)+1;
}
else{
printf("%d, ",i);
i=(i*2)+1;
}
}
gotoxy(12,15);
fflush(stdin);
r=random(3);
i=x;
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=(i*2)+1;

```

## Analysis of Computer programming tasks and Aptitudes

```

    if(i==(ans-3) || (i==ans))
    i=(i*2)+1;
    printf("%c)%d   ",ch[j],i);
    }
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                /*Question No. 7 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q7>Which one number will complete the number series ?");
i=random(10);
i=i+8;
gotoxy(12,12);
printf(" %d, %d, %d, %d, ___ ",i*i, (i-1)*(i-1), (i-2)*(i-2), (i-
3)*(i-3));
ans=(i-4)*(i-4);
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d   ",ch[j++],ans);
break;
}
i=random(ans);
if(i<26)
i++;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                /*Question No. 8 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q8>Which one number will complete the number series ?");
i=random(10);
i++;
gotoxy(12,12);
printf(" %d, %d, %d, %d, ___
",i*i, (i+1)*(i+1), (i+2)*(i+2), (i+3)*(i+3));
ans=(i+4)*(i+4);
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){

```

```

while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=random(ans);
if(i<56)
i++;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 9 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,8);
i=random(20);
i++;
printf("Q9> %d %d %d",i,i,i);
gotoxy(12,9);
printf(" %d %d %d",i+2,i+4,i+6);
gotoxy(12,10);
printf(" %d %d ?",i+3,i+6);
gotoxy(12,11);
printf("-----");
ans=i+9;
x=i*(i+2)*(i+3);
y=i*(i+4)*(i+6);
z=i*(i+6)*(i+9);
gotoxy(12,12);
printf(" %d %d %d",x,y,z);
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=random(30);
if(i==ans)
i++;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

```

## Analysis of Computer programming tasks and Aptitudes

```
        /*Question No. 10 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q10> Which number is wrong in the given series ?");
i=random(9);
i++;
x=i;
y=x+i;
gotoxy(12,12);
for(j=1;j<9;j++){
if(j==1){
printf("%d, ",y);
y=(y*2)-2;
j++;
}
if(j%2==0) {
printf("%d, ",x);
x=(x*2)+2;
}
else if(j==7) {
printf("%d, ",x+7);
ans=x+7;
} else {
printf("%d, ",y);
y=(y*2)-2;
}
}
gotoxy(12,14);
fflush(stdin);
r=random(3);
x=i;
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
x=(x*2)+2;
if(x==(ans-7) || (x==ans))
x=i;
printf("%c)%d    ",ch[j],x);
}
a=toupper(getche());
if(a==c)
result++;
printf("\nResult : %d",result);
getch();

        /*Question No. 11*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q11> Which set of numbers is not like the given set ?");
i=random(25);
```

```

if(i<13)
i=13;
x=i-2;
y=x;
gotoxy(12,12);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d, %d, %d ",ch[j++],i+213,i*5,i);
break;
}
printf("%c)%d, %d, %d ",ch[j],x*9,x*5,x);
x=x-3;
}
a=toupper(getche());
if(a==c)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 12*/

clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q12>Which one number will complete the number series ?");
i=random(10);
i++;
gotoxy(12,12);
x=3;
for(j=0;j<7;j++){
if(j==6){
ans=i;
printf("__");
break;
}
printf("%d ",i);
i=i+x;
x=x+4;
}
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);

```



```

getch();

                /*Question No. 13*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,8);
i=random(15);
if(i<7)
i=7;
printf("Q13> %d  %d  %d",i,i-4,i-5);
gotoxy(12,9);
printf("      %d  ?  %d",i-5,i-2);
gotoxy(12,10);
printf("      %d  %d  %d",i-2,i-4,i);
gotoxy(12,11);
printf("-----");
x=i*(i-5)*(i-2);
y=(i-4)*(i+1)*(i-4);
z=(i-5)*(i-2)*i;
ans=i+1;
gotoxy(12,12);
printf("  %d  %d  %d",x,y,z);
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
if(i==ans)
i++;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                /*Question No. 14*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q14 >Which number can be placed at the sign of
interrogation ?");
i=random(10);
i=i+2;
gotoxy(12,12);
for(j=0;j<5;j++){
if(j==4){
ans=i;
printf("__");
}
}

```

```

break;
}
printf("%d, ",i);
i=(i*2)-1;
}
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 15*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q15>Which one number will complete the number series ?");
i=random(20);
i++;
f=i/3.7;
if(f==0)
f++;
gotoxy(12,12);
printf(" %.01Ff, %.01Ff, %.01Ff,____ ",f,f-.5,f-1);
af=f-1.5;
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%0.01Ff  ",ch[j++],af);
break;
}
f=j;
f1=f/3.7;
f=af-f1;
printf("%c)%0.01Ff ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

```

## Analysis of Computer programming tasks and Aptitudes

```

                                /*Question No. 16*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q16>Which number comes next in the series ?");
i=random(10);
i++;
gotoxy(12,12);
printf(" %d,%d,%d, %d,%d,%d, __, %d,%d ",i,i-1,i-2,i*i,(i-1)*(i-1),
1),(i-2)*(i-2),(i-1)*(i-1)*(i-1),(i-2)*(i-2)*(i-2));
ans=i*i*i;
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                                /*Question No. 17*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q17>Which one number will complete the number series ?");
i=random(200);
if(i<100)
i=100+i;
f=i;
gotoxy(12,12);
for(j=0;j<5;j++){
if(j==4){
af=f;
printf("__");
break;
}
printf("%.2Ff, ",f);
f=(f-1)/2;
}
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%.2Ff  ",ch[j++],af);
break;
}
}

```

```

f=j;
f=af+((f+1)/1.7);
printf("%c%.2Ff ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No. 18*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q18>Which one number will complete the number series ?");
i=random(200);
if(i<100)
i=100+i;
gotoxy(12,12);
for(j=0;j<5;j++){
if(j==4){
ans=i;
printf("__");
break;
}
printf("%d, ",i);
i=i/3;
}
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d ",ch[j++],ans);
break;
}
i=ans-j-1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No. 19*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q19>Which number comes next in the series ?");
i=random(10);
i++;
gotoxy(12,12);
printf(" %d,%d,%d, %d,%d,__,%d ",i,i*(i-2),i-2,(i-
2)*(i+1),i+1,i+4);

```

```

ans=(i+1)*(i+4);
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.20*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,10);
printf("Q20>Which number comes next in the series ?");
i=random(10);
i++;
gotoxy(12,12);
printf(" %d,%d,%d, %d,%d,%d, __, %d,%d ",i,i-1,i-2,i*i,(i-1)*(i-1),
(i-2)*(i-2),(i-1)*(i-1)*(i-1),(i-2)*(i-2)*(i-2));
ans=i*i*i;
gotoxy(12,14);
r=random(3);
for(j=0;j<4;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

clrscr();
gotoxy(16,14);
printf("YOUR RESULT : %d\n\n",result);
printf("Thank you for your cooperation.");
getch();
return 0;

}

```

**NUMERICAL REASONING TEST**

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>
FILE *fp;
struct student{
char d;
char s_code[6];
char s_name[20];
char s_age[2];
char s_edu[15];
char s_ocu[15];
char inst_n[30];
char inst_a[40];
char s_add[40];
char s_exp[15];
char s_co[15];
char s_cd[20];
unsigned int s_res;
}S;

main(){
int i,j;
clrscr();
for(i=10;i<16;i++){
for(j=16;j<68;j++){
gotoxy(j,i);
printf("%c",220);
delay(5);
}
}
for(i=11;i<15;i++){
for(j=18;j<65;j++){
gotoxy(j,i);
printf("%c",255);
delay(5);
}
}
for(i=11;i<15;i++){
for(j=18;j<53;j++){
gotoxy(20,13);
printf(" WELCOME TO INDIAN STATISTICAL INSTITUTE ");
/*delay(20);*/
}
}
}
clrscr();
/*add();*/
clrscr();
gotoxy(30,4);
printf("NUMERICAL ABILITY");
gotoxy(31,7);
printf("DIRECTIONS");

```

## Analysis of Computer programming tasks and Aptitudes

```
gotoxy(10,10);
printf("This test consists of forty arithmetical problems.Next to
");
gotoxy(10,12);
printf("each problem there are five answers.These are marked by
A,");
gotoxy(10,14);
printf("B,C,D, and E.Only one of these suggested answer is right.");
gotoxy(10,16);
printf("You are to pick out the one correct answer for each
problem.");
gotoxy(10,18);
printf("Then write the answer mark from keybord.And press enter");
gotoxy(10,20);
printf("for next question.");
gotoxy(10,25);
printf("Press any key to continue....");
clrscr();
/*test();*/
exam();
return 0;
}

add() {
    FILE *fp;
    long int l;
    char ans='Y';
    fp=fopen("Student.txt","a");
    fseek(fp,0L,2);
    l=ftell(fp);
    l=l/(long int)sizeof(struct student);
    sprintf(S.s_code,"%04ld",++l);
    gotoxy(8,1);
    printf(" Please Fill up the Form bellow ");
    gotoxy(10,3);
    printf("Student No:%s",S.s_code);
    gotoxy(10,5);
printf("Enter student name:");
gets(S.s_name);
gotoxy(10,7);
printf("Enter student age:");
gets(S.s_age);
gotoxy(10,9);
printf("Enter student Education:");
gets(S.s_edu);
gotoxy(10,11);
printf("Enter student ocupation:");
gets(S.s_ocu);
gotoxy(10,13);
printf("Enter student institute name:");
gets(S.inst_n);
gotoxy(10,15);
printf("Enter student institute address:");
gets(S.inst_a);
gotoxy(10,17);
printf("Enter student Address:");
gets(S.s_add);
```

## Analysis of Computer programming tasks and Aptitudes

```
gotoxy(10,19);
printf("Enter student experience:");
gets(S.s_exp);
gotoxy(10,21);
printf("Enter the course name which you done:");
gets(S.s_co);
gotoxy(10,23);
printf("Enter name of certificate or diploma:");
gets(S.s_cd);
gotoxy(10,28);
printf("Press any key for continue(Press N for exit)....");
ans=toupper(getche());
if(ans!='N'){
fwrite(&S,sizeof(S),1,fp);
fclose(fp);
}
else
fclose(fp);
return 0;
}

/*TESTING PARTS*/
test(){
char a;
clrscr();
gotoxy(8,2);
printf("DEMO TEST :");
gotoxy(8,3);
printf("ADD");
gotoxy(14,3);
printf("25");
gotoxy(14,4);
printf("71");
gotoxy(14,5);
printf("----");
gotoxy(8,7);
printf("ANS:- A.76 B.96 C.106 D.116");
gotoxy(8,9);
printf("In example 96 is the correct answer.So you have just write
B.");
gotoxy(8,11);
printf("Please try again....");
gotoxy(8,14);
printf("TEST :");
gotoxy(8,16);
printf("ADD");
gotoxy(14,16);
printf("55");
gotoxy(14,17);
printf("77");
gotoxy(14,18);
printf("----");
gotoxy(14,20);
printf("ANS:- A.76 B.132 C.156 D.119");
a=toupper(getche());
if(a=='B')
exam();
```



## Analysis of Computer programming tasks and Aptitudes

```
else {
clrscr();
printf("Sorry You Not Passed In Test...Please Try Later ...!");
}
return 0;
}

/* EXAM PARTS */
exam()
{
int s,k,r,i,j,result=0 ;
long ans;
float f,f1;
double af;
char c,a,ch[5]={'A','B','C','D','E'};
randomize();

/* Question No.1 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,6);
printf("Q1>ADD : ");
gotoxy(10,8);
for(j=0;j<5;j++){
i=random(500);
i++;
printf("\n %d",i);
ans=ans+i;
}
gotoxy(8,20);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d ",ch[j++],ans);
break;
}
i=j+1;
i=ans+i;
printf("%c)%d ",ch[j],i);
}
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/* Question No. 2 */
clrscr();
ans=0;
gotoxy(12,12);
i=random(8000);
if(i<3000)
i=3000+i;
printf("Q2>SUBTRACT: %d",i);
ans=i;
```

```

gotoxy(30,13);
i=random(2000);
if(i<1000)
i=1000+(i/4);
printf("%d",i);
ans=ans-i;
gotoxy(30,14);
printf("----");
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=j+1;
i=i*10;
i=ans+i;
printf("%c)%d    ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

    /*Question No. 3*/

clrscr();
fflush(stdin);
gotoxy(12,12);
i=random(10);
i++;
f=i/2.3;
printf("Q3>MULTIPLY:      %.2Ff",f);
gotoxy(30,13);
f1=i/9.3;
printf("%.2Ff",f1);
gotoxy(30,14);
printf("-----");
gotoxy(8,16);
af=f*f1;
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%2Ff    ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/.3);
printf("%c)%2Ff    ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());

```

```

if(c==a)
result++;
printf("\nResult : %d",result);
getch();

        /* Question No. 4 */

clrscr();
fflush(stdin);
ans=0;
gotoxy(12,12);
i=random(25);
i++;
f=i/2.3;
printf("Q4>MULTIPLY:          %.2Ff", f);
gotoxy(30,13);
f1=i/3.3;
printf("%.2Ff", f1);
gotoxy(30,14);
printf("-----");
af=f*f1;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.2Ff      ",ch[j++],af);
break;
}
i=j+1;
f=af+(i/.3);
printf("%c)%.2Ff      ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

        /*Question No. 5 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,12);
i=random(180);
if(i<100)
i=113;
printf("Q5>MULTIPLY:          %d",i);
ans=i;
gotoxy(30,13);
i=random(150);
if(i<100)
i=129;
printf("%d",i);
ans=ans*i;
gotoxy(30,14);

```

```

printf("----");
k=1;
gotoxy(8,16);
fflush(stdin);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d    ",ch[j++],ans);
break;
}
i=k*10;
i=ans+i;
printf("%c)%d    ",ch[j],i);
k=k+1;
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 6 */
clrscr();
fflush(stdin);
ans=0;
gotoxy(12,12);
i=random(10);
i++;
f=i/5.3;
printf("Q6>MULTIPLY:      %.2Ff",f);
af=f*f;
gotoxy(30,13);
printf("%.2Ff",f);
gotoxy(30,14);
printf("-----");
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.2Ff    ",ch[j++],af);
break;
}
f=j+2;
f=af+(f/.3);
printf("%c)%.2Ff    ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 7 */

```

```

clrscr();
fflush(stdin);
ans=0;
gotoxy(12,12);
i=random(5);
i++;
f=i;
f=f/11.33;
printf("Q7>MULTIPLY:          %.2Ff", f);
af=f*f;
gotoxy(30,13);
printf("%.2Ff", f);
gotoxy(30,14);
printf("-----");
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.3Ff      ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/5.33);
printf("%c)%.3Ff      ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 8 */
clrscr();
fflush(stdin);
gotoxy(12,12);
i=random(60);
i++;
printf("Q8>DIVIDE:          %d/%d",i,i+23);
f=i;
f1=i+23;
af=f/f1;
gotoxy(12,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.3Ff      ",ch[j++],af);
break;
}
i=j+1;
f=af+(i/8.3);
printf("%c)%.3Ff      ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());

```

## Analysis of Computer programming tasks and Aptitudes

```
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 9 */
clrscr();
ans=0;
gotoxy(12,12);
i=random(10);
i++;
f=i/5.7;
f1=i/1.3;
printf("Q9>DIVIDE:          %.2Ff / %.2Ff ",f,f1);
af=f/f1;
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.2Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/8.3);
printf("%c)%.2Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 10 */
clrscr();
af=0;
gotoxy(12,12);
i=random(10);
i++;
f=i/2.5;
f1=i/4.5;
printf("Q10>DIVIDE:          %.2Ff / %.2Ff ",f,f1);
af=f/f1;
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.3Ff  ",ch[j++],af);
break;
}
i=j+1;
f=af+(i/7.4);
printf("%c)%.3Ff  ",ch[j],f);
}
gotoxy(12,20);
```

## Analysis of Computer programming tasks and Aptitudes

```
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 11*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(200);
i++;
f=i/7.7;
f1=i/1.3;
printf("Q11>DIVIDE:          %1.2Ff / %2.2Ff ",f,f1);
af=f/f1;
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%0.2Ff      ",ch[j++],af);
break;
}
i=j+1;
f=af+(i/5.3);
printf("%c)%0.2Ff      ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 12*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(20);
i++;
f=i/11.7;
f1=i/3.7;
printf("Q12>DIVIDE:          %.2Ff / %.2Ff ",f,f1);
af=f/f1;
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%0.2Ff      ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/7.3);
printf("%c)%0.2Ff      ",ch[j],f);
}
}
```

## Analysis of Computer programming tasks and Aptitudes

```
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 13*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(20);
i++;
f=i;
f1=f*2;
printf("Q13> 1/%.0Ff / 1/%.0Ff ",f,f1);
af=(1/f)/(1/f1);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.2Ff ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/3.3);
printf("%c)%.2Ff ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 14*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(15);
if(i<9)
i=9;
printf("Q14> %d/%d X %d/%d ",i-3,i,i+1,i+2);
k=(i-3)*(i+1);
ans=i*(i+2);
gotoxy(8,14);
f1=k;
af=ans;
af=f1/af;
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c) %.2Ff ",ch[j++],af);
break;
}
}
```



## Analysis of Computer programming tasks and Aptitudes

```

f=j+1;
f=af+(f/9.7);
printf("%c) %.2Ff    ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                /*Question No. 15*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(20);
if(i<15)
i=13;
printf("Q15>    %d/%d X %d/%d ",i,i-5,i+1,i-3);
k=i*(i+1);
ans=(i-5)*(i-3);
f=k;
af=ans;
af=(f/af);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c) %.2Ff    ",ch[j++],af);
break;
}
}
f=j+1;
f=af+(f/3.7);
printf("%c) %.2Ff    ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

                /*Question No. 16*/
clrscr();
fflush(stdin);
af=0;
gotoxy(10,6);
printf("Q16>ADD :          ");
gotoxy(10,8);
i=random(9);
i++;
printf("\n          %d/%d",1,i);
f=i;
af=af+(1/f);
printf("\n          %d/%d",i+1,i+2);
af=af+((f+1)/(f+f));

```

```

printf("\n      %d/%d",1,i*4+1);
af=af+(1/(f*4));
gotoxy(8,20);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").2Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/7.3);
printf("%c").2Ff  ",ch[j],f);
}
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 17*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(10);
k=random(15);
i++;
if(k<10)
k=i+k;
printf("Q17>ADD:      %dft. %din.",i,k);
ans=i;
s= k;
gotoxy(25,13);
i=random(10);
k=random(15);
i++;
if(k<10)
k=i+k;
printf("%dft. %din.",i,k);
ans=ans+i;
s= s+k;
gotoxy(25,14);
i=random(10);
k=random(13);
i++;
if(k<8)
k=i+k;
printf("%dft. %din.",i,k);
ans=ans+i;
s= s+k;
gotoxy(2,18);
while(s>12){
ans++;
s=s-12;
}
r=random(4);
for(j=0;j<5;j++){

```

```

while(j==r){
c=ch[j];
printf(" %c)%dft.",ch[j++],ans);
printf(" %din.",s);
break;
}
i=ans+j+2;
k=s-j+2;
printf("%c) %dft. %din.",ch[j],i,k);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No. 18*/
clrscr();
ans=0;
gotoxy(12,12);
i=random(50);
k=random(1000);
i++;
if(k<400)
k=445+i;
printf("Q18>ADD:      %dKg. %d gm.",i,k);
ans=i;
s= k;
gotoxy(25,13);
i=random(60);
k=random(1000);
i++;
if(k<400)
k=445+i;
printf("%dKg. %d gm.",i,k);
ans=ans+i;
s= s+k;
gotoxy(25,14);
i=random(40);
k=random(999);
i++;
if(k<400)
k=445+i;
printf("%dKg. %d gm.",i,k);
ans=ans+i;
s= s+k;
gotoxy(2,18);
if(s>1000){
ans++;
s=s-1000;
}
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%dKg.",ch[j++],ans);

```

```

printf(" %dgm.",s);
break;
}
i=ans+(j*7);
printf("%c)%dKg. %dgm. ",ch[j],i,k);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No. 19*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(20);
if(i<12)
i=12;
ans=i++;
i=ans*ans;
printf("Q19>Square root: %d",i);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d ",ch[j],i);
}
gotoxy(12,16);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.20*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(10);
i++;
f=i/1.3;
f1=f*f;
printf("Q20>Square root: %.02Ff",f1);
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%.3Ff ",ch[j++],f);
break;
}
}

```

```

f1=j+1;
f1=f+(f1/7.3);
printf("%c%.3Ff  ",ch[j],f1);
}
gotoxy(12,16);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.21*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(10);
i++;
f=i/2.1;
af=f*f;
printf("Q21>Square root:  %.02Ff",af);
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c%.02Ff  ",ch[j++],f);
break;
}
f1=j+1;
f1=f+f1/7.9;
printf("%c%.02Ff ",ch[j],f1);
}
gotoxy(12,16);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.22*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(100);
if(i<9)
i=9+i;
printf("Q22>  ?= %d/%d X 12%",i,i-9);
f=i;
f1=i-9;
af=(f/f1)*.12;
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c%.02Ff  ",ch[j++],af);
break;
}
}

```

```

f=j+1;
f=af+(f/33.34);
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.23*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(200);
if(i<15)
i=16+i;
printf("Q23>  ?= %d/%d X 13%",i,i-13);
f=i;
f1=i-13;
af=(f/f1)*.13;
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
}
f=j+1;
f=af+f/33.4;
if(f==af)
f++;
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.24*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(20);
if(i<5)
i=5+i;
printf("Q24>  ?= 727 X %d/%d % ",i,i-3);
f=i;
f1=i-3;
af=727*((f/f1)/100);
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){

```

```

while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
i=j+1;
f=af+i/.4;
if(f==af)
f++;
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.25*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(100);
if(i<15)
i=15+i;
printf("Q25>  %d = ? X %d %",i,i-11);
f=i;
af=(f*100)/(f-11);
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").3Ff  ",ch[j++],af);
break;
}
f=j+2;
f=af+f;
printf("%c").3Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.26*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(200);
i++;
af=i*.2;
printf("Q26>  %.02Ff = %d X ?%",af,i);
gotoxy(8,15);
r=random(4);

```

## Analysis of Computer programming tasks and Aptitudes

```

for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],20);
break;
}
i=j+1;
i=20+i;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.27*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,12);
i=random(80);
if((i<25)|| (i==50))
i=26;
af=i*.25;
printf("Q27>  %.02Ff = %d X ?%",af,i);
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],25);
break;
}
i=(j*2)+1;
i=25+i;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.28*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(20);
i++;
printf("Q28>  %d/%d = ?/%d",i,i+2,i*3);
f=i;
af=(f*(f*3))/(f+2);
gotoxy(8,15);
r=random(4);

```



```

for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+f;
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.29*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(7);
i++;
printf("Q29>  %d/%d = %d/?",i,i*7,i*13);
f=i;
af=((f*7)*(f*13))/f;
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/2.3);
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.30*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(25);
if(i<15)
i=13;
printf("Q30>  %d/%d = %d/?",i,i-9,i*3);
f=i;
af=((f-9)*(f*3))/i;
gotoxy(8,15);
r=random(4);
for(j=0;j<5;j++){

```

```

while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/1.3);
printf("%c").02Ff  ",ch[j],f);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.31*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(10);
i++;
printf("Q31>Which one number can replace both question marks ?");
gotoxy(12,14);
printf(" %d/? = ?/%d",i,i*100);
ans=i*10;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=j+1;
i=ans+i;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.32*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(20);
i++;
printf("Q32>Which one number can replace both question marks ?");
gotoxy(12,14);
printf(" %d/? = ?/%d",i,i*81);
ans=i*9;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){

```

```

while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=j+1;
i=ans+i;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.33*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(30);
i++;
printf("Q33>Which one number can replace both question marks ?");
gotoxy(12,14);
printf(" %d/? = ?/%d",i,i*49);
ans=i*7;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=random(100);
if(i==ans);
i++;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.34*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(30);
i++;
printf("Q34>Which one number can replace both question marks ?");
gotoxy(12,14);
printf(" %d/? = ?/%d",i,i*121);
ans=i*11;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){

```

```

while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.35*/
clrscr();
ans=0;
gotoxy(10,12);
i=random(15);
i++;
printf("Q35>Which one number can replace both question marks ?");
gotoxy(12,14);
printf(" %d/? = ?/%d",i,i*169);
ans=i*13;
gotoxy(8,16);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d  ",ch[j],i);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.36*/
clrscr();
fflush(stdin);
ans=0;
gotoxy(10,10);
i=random(200);
i++;
printf("Q36> List price = Rs. %d",i);
gotoxy(10,12);
printf("First Discount = 20 %");
af=i-(i*.2);
gotoxy(10,14);
printf("Further Discount = 2 %");
af=af-(af*.02);
gotoxy(10,16);
printf("Net Price = ?");

```

```

gotoxy(8,18);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
f1=j+1;
f=af+(f1/7.5);
printf("%c").02Ff  ",ch[j],f);
fflush(stdin);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.37*/

clrscr();
fflush(stdin);
ans=0;
gotoxy(10,10);
i=random(200);
i++;
printf("Q37> List price = Rs. %d",i);
gotoxy(10,12);
printf("First Discount = 25 %");
af=i-(i*.25);
gotoxy(10,14);
printf("Further Discount = 5 %");
af=af-(af*.05);
gotoxy(10,16);
printf("Net Price = ?");
gotoxy(8,18);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c").02Ff  ",ch[j++],af);
break;
}
f=j+1;
f=af+(f/1.2);
printf("%c").02Ff  ",ch[j],f);
fflush(stdin);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();

/*Question No.38*/

clrscr();
gotoxy(10,10);

```

```

ans=0;
i=random(10);
i++;
k=random(10);
k++;
ans=i*k;
i=i*i*i;
k=k*k*k;
printf("Q38> Cube root %d X %d",i,k);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d  ",ch[j],i);
fflush(stdin);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
/*Question No.39*/
clrscr();
gotoxy(10,10);
ans=0;
i=random(25);
if(i<11)
i=13;
k=random(25);
if(k<11)
k=11;
ans=i*k;
i=i*i*i;
k=k*k*k;
printf("Q39> Cube root %d X %d",i,k);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d  ",ch[j],i);
fflush(stdin);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;

```

```

printf("\nResult : %d",result);
getch();
        /*Question No.40*/
clrscr();
gotoxy(10,10);
ans=0;
i=random(15);
if((i==0)|| (i==10))
i++;
k=random(15);
if((k==0)|| (k==10))
k++;
ans=i*k;
i=i*i*i;
k=k*k*k;
printf("Q40>  Cube root %d X %d",i,k);
gotoxy(8,14);
r=random(4);
for(j=0;j<5;j++){
while(j==r){
c=ch[j];
printf("%c)%d  ",ch[j++],ans);
break;
}
i=ans+j+1;
printf("%c)%d  ",ch[j],i);
fflush(stdin);
}
gotoxy(12,20);
a=toupper(getche());
if(c==a)
result++;
printf("\nResult : %d",result);
getch();
clrscr();
gotoxy(16,14);
printf("YOUR RESULT : %d\n\n",result);
printf("Thank you for your cooperation.");
getch();
return 0;
}

```

1

91

**SHORT TERM MEMORY TEST**

```

import java.awt.*;
import java.awt.event.*;
import java.awt.Color.*;
import java.lang.*;
import java.lang.String;
import java.lang.Integer;
import java.io.File;
import java.io.*;

public class test1 extends Frame implements
ActionListener,KeyListener {

    TextField inum;

    private Font font;
    long n,dgt=10,m=1000;
    int
trial=1,i,span=9,spanl=9,time,time1,n1,level=2,mm,nn,row=-1,rowl=-
1,now_row,now_rowl,lv=6,lv1=6,chance=-1,chancel=-1,num2=0,num3=0;

    static File f;
    FileOutputStream fout;
    boolean
c,c1,d=false,d1=false,wrong=true,wrongl=true,key=true,keyl=true,fwri
te=false,fwritel=false,fhead=true,spn=true,spnl=true,
    instruction = true , name = true , tr2 = false , con
= true , demo_trial = true , dr = false;
    static boolean rl=true;
    String
nm,bb,ss,bb1,ss1,sphead="SPAN=",sp="_____" ,cc,ccl,nl="\n",dt="Dsp.T
m ",nd="Nm.Dsp ",nr="Nm.Rsp ",
    dm,tot="Tot.RT ",rt="Reac.T
",md="Mid.Prd",dtime,rtime,ktime,mdtime;
    String rules[]={
        "PLEASE READ THE FOLLOWING INSTRUCTIONS BEFORE
YOU START :",
        "
        _____",
        "
        ",
        "1. Input your name at the textbox through
keyboard. ",
        "2. Press 'ENTER' key to clear the screen
",
        "3. Again press 'ENTER' key , immediately a
number will ",
        " be displayed.
",
        "4. Remember the number and input it at the
box. ",
        "5. Immediately another number will be
displayed. ",
        "6. Remember the number and input it.
",
    }

```



## Analysis of Computer programming tasks and Aptitudes

```

of the test.",
    "7. The process will be repeated up to the end
of the test.",

};

long
t1,t2,t3,t4,pt1,tt1,tt2,tt3,tt4,ptt1,number_display=0,number_display
l=0,previous_number=0;
double display_time,respond_time,keypress_time,mid_period;
long ar[] =
{582,6439,42731,619473,5917428,58192647,275862584};
long cr[] =
{694,7286,75836,392487,4179386,38295174,713942568};
long br[][] = {
    {275      ,816      ,531      ,857      ,382
,631      ,941      ,683      ,854      ,524      ,351      ,912
,854      ,163      ,794},
    {2417     ,5879     ,4618     ,5283
,6319     ,7195     ,8473     ,9715     ,2483     ,5726     ,4197
,5192     ,1684     ,8526     ,5714},
    {42859    ,16253    ,59317    ,82631
,75183    ,82764    ,71826    ,39724    ,29856    ,48729    ,72631
,26195    ,36491    ,79432    ,13846},
    {314986   ,529671   ,637148   ,758693
,847136   ,938561   ,357428   ,243569   ,735291   ,613594   ,194625
,746253   ,634918   ,413827   ,914263},
    {5369718  ,6857493  ,7586942  ,8697514
,9584721  ,6385427  ,2143756  ,3852197  ,4351829  ,7461392  ,6413579
,3971862  ,4792638  ,5973281  ,4698537},
    {63579182 ,72859463 ,86475392 ,14283796
,24136957 ,39264158 ,52384169 ,53748921 ,61429358 ,73964152
,93851426 ,93825147 ,42719586 ,35716428 ,94637521},

{285946371,427316859,312486975,815274639,471638592,812746935,9516384
27,371495268,741369285,625749318,736594182,685147293,135248697,82571
4973,741625938}
};

public static void main(String args[] ){
    f = new File("a:\\\" + args[0]);
    test1 ff=new test1("test1");
    ff.setSize(800,800);
    ff.setVisible(true);

}

public test1(String s) {
    super(s);
    setLayout(new FlowLayout());
    font = new Font("serif",Font.BOLD,20);

    try{
        fout = new FileOutputStream(f);
    } catch(IOException
e){System.out.println("IOERRR");}

    this.setFont(font);
}

```

## Analysis of Computer programming tasks and Aptitudes

```
        setBackground(Color.black);
        setForeground(Color.red);

        // Label in_label = new Label("Input number
",Label.RIGHT);

        inum = new TextField(12);

        this.add(inum);
        this.inum.requestFocus();

        // this.add(in_label);

        inum.addActionListener(this);
        inum.addKeyListener(this);
        mm = (int) (Math.random()*1000);
        nn = mm%5;
    }

    public void writefile() throws IOException {
        if(fhead) {
            try {
                fout.write(dt.getBytes());
                fout.write(nd.getBytes());
                fout.write(nr.getBytes());
                fout.write(tot.getBytes());
                fout.write(rt.getBytes());
                fout.write(md.getBytes());
                fout.write(nl.getBytes());

            } catch (IOException ex)
{System.out.println("ERRRR");}

            }
            fhead=false;

            if(trial==1)
            {
                try {

                    fout.write(dtime.getBytes());
                    fout.write(sp.getBytes());
                    fout.write(bb.getBytes());
                    fout.write(sp.getBytes());
                    fout.write(cc.getBytes());
                    fout.write(sp.getBytes());
                    fout.write(rtime.getBytes());
                    fout.write(sp.getBytes());
                    fout.write(ktime.getBytes());
                    fout.write(sp.getBytes());
                    fout.write(mdtype.getBytes());

                } catch (IOException ex)
{System.out.println("ERRRR");}
            }
        }
```

## Analysis of Computer programming tasks and Aptitudes

```
        if(trial==2)
        {
            try {

                fout.write(dtime.getBytes());
                fout.write(sp.getBytes());
                fout.write(bb1.getBytes());
                fout.write(sp.getBytes());
                fout.write(cc1.getBytes());
                fout.write(sp.getBytes());
                fout.write(rtime.getBytes());
                fout.write(sp.getBytes());
                fout.write(ktime.getBytes());
                fout.write(sp.getBytes());
                fout.write(mdtype.getBytes());

            } catch (IOException ex)
        {System.out.println("ERRRR");}
        }

    }

////////////////////////////////////

    public void keyPressed(KeyEvent kee) { }

    public void keyTyped(KeyEvent ke)
    {
        if(!demo_trial)
        {
            if(trial==1)
            {
                if(key)
                t2=System.currentTimeMillis();
                key=false;
            }
            if(trial==2)
            {
                if(key1)
                tt2=System.currentTimeMillis();
                key1=false;
            }
        }
    }

    public void keyReleased(KeyEvent k) { }

////////////////////////////////////

    public void actionPerformed(ActionEvent ae) {
```

## Analysis of Computer programming tasks and Aptitudes

```
if(name)
{
    nm = inum.getText();
    try {
        fout.write(nm.getBytes());
        fout.write(nl.getBytes());
        name = false;
    } catch(IOException ex) { }
}

if(instruction)
    repaint();

if(!instruction)
{
    if(demo_trial)
    {
        if(dr)
            dm = inum.getText();
        repaint();
    }

    if(!demo_trial)
    {

        /////      TRIAL 1 STARTS HERE      /////

        if(trial==1)
        {
            row+=1;
            c=true;
            key=true;

            cc=inum.getText();

            if(row<7)
            {
                number_display = ar[row];
                if(d)

t3=System.currentTimeMillis();

            }

            if(d &&
!(bb.equals(inum.getText())))
            {
                if(row>=7)
                    chance+=1;

                if(spn && row<=7)
                {
```

## Analysis of Computer programming tasks and Aptitudes

```
        span = row+1;
        spn = false;
    }
    if(wrong)
        lv=row-2;
    wrong = false;
}
if(row>=7 && chance<2 && num2<15)
{
    if(lv >= 0 && lv <= 6)
        number_display =

    else
        number_display =

        num2 += 1;
        if(num2 > 14)
            num2 -= 14;
        if(d)

br[lv][num2];

br[now_row][num2];

t3=System.currentTimeMillis();
    }

    repaint();
}

/////    TRIAL 1 ENDS HERE    /////
/////    TRIAL 2 STARTS HERE    /////

if(trial==2)
{
    tr2 = true;
    row1 += 1;
    c1=true;
    key1=true;

    cc1=inum.getText().trim();

    if(row1<7)
    {
        number_display1 = cr[row1];
        if(d1)

tt3=System.currentTimeMillis();
    }

    if(d1 && !(bb1.equals(cc1)))
    {
        if(row1>=7)
            chance1+=1;

        if(spn1 && row1<=7)
        {
```

## Analysis of Computer programming tasks and Aptitudes

```

        span1 = row1+1;
        spn1 = false;
    }

    if(wrong1)
        lv1=row1-2;

    wrong1 = false;
}

if(row1>=7 && chance1<2 && num3<15)
{
    if(lv1 >= 0 && lv1 <= 6)
        number_display1 =

br[lv1][num3];

    else
        number_display1 =

br[now_row1][num3];

    num3 += 1;
    if(num3 > 14)
        num3 -= 14;
    if(d1)

tt3=System.currentTimeMillis();

}

// cc1 = inum.getText();

repaint();

}

//////// TRIAL 2 ENDS HERE //////////

} //// end of if(!demo_trial) loop
} //// end of if(!instruction) loop
instruction = false;
} ////end of ActionPerformed()

public void focusPerformed(FocusEvent fe) { }

////////////////////////////////////
////////////////////////////////////
// ***** PAINT *****//
////////////////////////////////////
////////////////////////////////////

```

## Analysis of Computer programming tasks and Aptitudes

```

public void paint(Graphics g) {

    if(instruction)
    {
        for(i=0;i<11;i++)
            g.drawString(rules[i],100,150+i*30);
    }
    if(!instruction)
    {
        if(demo_trial)
        {

            g.drawString("* DEMO TRIAL *",325,80);
            g.setColor(Color.white);
            g.drawString("INPUT THE NUMBER DISPLAYED IN
THE BOX",200,100);

            g.setColor(Color.red);
            if(!dr)
            {
                g.drawString("738",350,325);
                try {
                    Thread.sleep(2000);
                } catch(Exception e) { }
                g.drawRect(0,150,1000,600);
                g.setColor(getBackground());
                g.fillRect(0,150,1000,600);
                inum.setText("");

            }
            if(dr)
            {
                if(dm.equals("738"))
                    g.drawString("! YOU ANSWER IS CORRECT
!",250,350);

                else
                    g.drawString("! YOU ANSWER IS WRONG
!",250,350);

                demo_trial = false;
            }
            dr = true;
        }

        if(!demo_trial)
        {

            if(con) {
                g.setColor(Color.white);
                g.drawString("PRESS 'ENTER' TO
CONTINUE",250,400);

                g.setColor(Color.red);
                con = false;
            }
            if(trial==1)
            {
                if(c && chance<2)
                {
                    try {

```

## Analysis of Computer programming tasks and Aptitudes

```
g.drawString(String.valueOf(number_display), 350, 325);

        if(d) {
            now_row = row-1;
            if(now_row>6)
                now_row = 6;
        }

inum.setText("");

if(row>=7) {

Thread.sleep(2000-time);

(2000-time)/1000.0;

        display_time =
            if(time<1900)
                time+=100;
            }
        else {
            Thread.sleep(2000);
            display_time = 2;
        }
        g.drawRect(0, 150, 1000, 600);
        g.setColor(getBackground());
        g.fillRect(0, 150, 1000, 600);
        //

g.setColor(getForeground());

t1=System.currentTimeMillis();

        d=true;

dtime=String.valueOf(display_time);

        if(d) {
            respond_time = (t3 -
            pt1)/1000.0;
            keypress_time = (t2
            - pt1)/1000.0;
            mid_period =
            (respond_time - keypress_time)/1.0;

rtime=String.valueOf(respond_time);
ktime=String.valueOf(keypress_time);
mdtime=String.valueOf(mid_period);

        }

pt1=t1;
```



## Analysis of Computer programming tasks and Aptitudes

```

e) { }

} catch(InterruptedExcepcion

if(fwrite & chance<2)
    try {
        writefile();

fout.write(nl.getBytes());
    } catch (IOException
io) {System.out.println("ERRRR");}

fwrite = true;

bb=String.valueOf(number_display);

} //end if

else if(c && chance>=2)
{
g.setColor(Color.white);
ss = String.valueOf(span);
g.drawString("span:" +

ss,180,320);
g.drawString("PRESS 'ENTER'
TO CONTINUE",180,350);

trial += 1;
g.setColor(Color.red);
try{

fout.write(nl.getBytes());
fout.write(nl.getBytes());
fout.write(sphead.getBytes());
fout.write(ss.getBytes());
fout.write(nl.getBytes());
fout.write(nl.getBytes());

} catch(IOException e)
{System.out.println("span not printed");}

inum.setText("");
}

// g.drawString("Chance: " +
chance,6,320);

}

```

## Analysis of Computer programming tasks and Aptitudes

```
////////////////////////////////////
////////////////////////////////////

        if(tr2 && trial==2)

            {

                if(c1 && chance1<2)
                    {
                        try
                        {

g.drawString(String.valueOf(number_display1),350,325);

                                if(d1) {
                                    now_row1 = row1-1;
                                    if(now_row1>6)
                                        now_row1 = 6;

                                    }

                                inum.setText(" ");

                                if(row1>=7)
                                    {
                                        Thread.sleep(2000-
time1);
                                        display_time =
(2000-time1)/1000.0;

                                        if(time1<1900)
                                            time1 += 100;
                                        }
                                    else
                                        {
                                            Thread.sleep(2000);
                                            display_time = 2;
                                        }
                                g.drawRect(0,150,1000,600);
                                g.setColor(getBackground());
                                g.fillRect(0,150,1000,600);
                                g.setColor(getForeground());

                                tt1=System.currentTimeMillis();

                                d1=true;

                                dtime=String.valueOf(display_time);

                                if(d1) {
                                    respond_time = (tt3
- ptt1)/1000.0;
                                }

```

## Analysis of Computer programming tasks and Aptitudes

```

- ptt1)/1000.0;
(respond_time - keypress_time)/1.0;

rtime=String.valueOf(respond_time);
ktime=String.valueOf(keypress_time);
mdtime=String.valueOf(mid_period);

        }
        ptt1 = ttl;

    } catch (InterruptedException
e) { System.out.println("TRIAL2 ERR"); }

        if (fwritel && chance1 < 2)
            try {
                writefile();

        } catch
(IOException io) { System.out.println("ERRRR"); }

        fwritel = true;

        bb1 =

    } //end if

    else if (c1 && chance1 >= 2)
        {

g.setColor(Color.white);
g.drawString("*****THANK YOU*****", 180, 290);
        ss1 =
String.valueOf(span1);
        g.drawString("span:"
+ ss1, 180, 320);

        try
        {

fout.write(nl.getBytes());
fout.write(nl.getBytes());
fout.write(sphead.getBytes());
fout.write(ss1.getBytes());

```

## Analysis of Computer programming tasks and Aptitudes

```
        }
catch(IOException e) {System.out.println("span not printed");}
        inum.setText("");
    }

    //        g.drawString("Chance: " + chance1,6,320);
        }
        ////// TRIAL 2 ENDS HERE ////
    // trial += 1;
        }
    }
    // instruction = false;
} //end paint
}
```