Year 2019 Examination

ISC

# Analysis of Pupil Performance

## COMPUTER SCIENCE

Research Development and Consultancy Division

Council for the Indian School Certificate Examinations

New Delhi

**Year 2019**

---

***Published by*:**

Research Development and Consultancy Division (RDCD)

Council for the Indian School Certificate Examinations

Pragati House, 3rd Floor

47-48, Nehru Place

New Delhi-110019

Tel: (011) 26413820/26411706
E-mail: council@cisce.org

# FOREWORD

This document of the Analysis of Pupils' Performance at the ISC Year 12 and ICSE Year 10 Examination is one of its kind. It has grown and evolved over the years to provide feedback to schools in terms of the strengths and weaknesses of the candidates in handling the examinations.

We commend the work of Mrs. Shilpi Gupta (Deputy Head) of the Research Development and Consultancy Division (RDCD) of the Council and her team, who have painstakingly prepared this analysis. We are grateful to the examiners who have contributed through their comments on the performance of the candidates under examination as well as for their suggestions to teachers and students for the effective transaction of the syllabus.

We hope the schools will find this document useful. We invite comments from schools on its utility and quality.

**Gerry Arathoon**
**Chief Executive & Secretary**

**October 2019**

# PREFACE

The Council has been involved in the preparation of the ICSE and ISC Analysis of Pupil Performance documents since the year 1994. Over these years, these documents have facilitated the teaching-learning process by providing subject/ paper wise feedback to teachers regarding performance of students at the ICSE and ISC Examinations. With the aim of ensuring wider accessibility to all stakeholders, from the year 2014, the ICSE and the ISC documents have been made available on the Council's website www.cisce.org.

The documents include a detailed qualitative analysis of the performance of students in different subjects which comprises of examiners' comments on common errors made by candidates, topics found difficult or confusing, marking scheme for each question and suggestions for teachers/ candidates.

In addition to a detailed qualitative analysis, the Analysis of Pupil Performance documents for the Examination Year 2019 also have a component of a detailed quantitative analysis. For each subject dealt with in the document, both at the ICSE and the ISC levels, a detailed statistical analysis has been done, which has been presented in a simple user-friendly manner.

It is hoped that this document will not only enable teachers to understand how their students have performed with respect to other students who appeared for the ICSE/ISC Year 2019 Examinations, but also provide information on how they have performed within the Region or State, their performance as compared to other Regions or States, etc. It will also help develop a better understanding of the assessment/ evaluation process. This will help teachers in guiding their students more effectively and comprehensively so that students prepare for the ICSE/ ISC Examinations, with a better understanding of what is required from them.

The Analysis of Pupil Performance document for ICSE for the Examination Year 2019 covers the following subjects: English (English Language, Literature in English), Hindi, History, Civics and Geography (History and Civics, Geography), Mathematics, Science (Physics, Chemistry, Biology), Commercial Studies, Economics, Computer Applications, Economic Applications, Commercial Applications.

Subjects covered in the ISC Analysis of Pupil Performance document for the Year 2019 include English (English Language and Literature in English), Hindi, Elective English, Physics (Theory), Chemistry (Theory), Biology (Theory), Mathematics, Computer Science, History, Political Science, Geography, Sociology, Psychology, Economics, Commerce, Accounts and Business Studies.

I would like to acknowledge the contribution of all the ICSE and the ISC examiners who have been an integral part of this exercise, whose valuable inputs have helped put this document together.

I would also like to thank the RDCD team of Dr. M.K. Gandhi, Dr. Manika Sharma, Mrs. Roshni George and Mrs. Mansi Guleria who have done a commendable job in preparing this document.

***Shilpi Gupta***
***Deputy Head - RDCD***

***October 2019***

# CONTENTS

# INTRODUCTION

This document aims to provide a comprehensive picture of the performance of candidates in the subject. It comprises of two sections, which provide Quantitative and Qualitative analysis results in terms of performance of candidates in the subject for the ISC Year 2019 Examination. The details of the Quantitative and the Qualitative analysis are given below.

## Quantitative Analysis

This section provides a detailed statistical analysis of the following:

- Overall Performance of candidates in the subject (Statistics at a Glance)
- State wise Performance of Candidates
- Gender wise comparison of Overall Performance
- Region wise comparison of Performance
- Comparison of Region wise performance on the basis of Gender
- Comparison of performance in different Mark Ranges and comparison on the basis of Gender for the top and bottom ranges
- Comparison of performance in different Grade categories and comparison on the basis of Gender for the top and bottom grades

The data has been presented in the form of means, frequencies and bar graphs.

### Understanding the tables

Each of the comparison tables shows N (Number of candidates), Mean Marks obtained, Standard Errors and t-values with the level of significance. For t-test, mean values compared with their standard errors indicate whether an observed difference is likely to be a true difference or whether it has occurred by chance. The t-test has been applied using a confidence level of 95%, which means that if a difference is marked as ‘statistically significant’ (with * mark, refer to t-value column of the table), the probability of the difference occurring by chance is less than 5%. In other words, we are 95% confident that the difference between the two values is true.

t-test has been used to observe significant differences in the performance of boys and girls, gender wise differences within regions (North, East, South and West), gender wise differences within marks ranges (Top and bottom ranges) and gender wise differences within grades awarded (Grade 1 and Grade 9) at the ISC Year 2019 Examination.

The analysed data has been depicted in a simple and user-friendly manner.

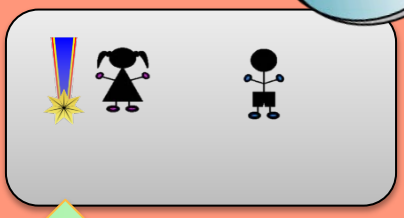
Given below is an example showing the comparison tables used in this section and the manner in which they should be interpreted.

**Comparison on the basis of Gender**

| Gender | N | Mean | SE | t-value |
|---|---|---|---|---|
| Girls | 2,538 | 66.1 | 0.29 | 11.91* |
| Boys | 1,051 | 60.1 | 0.42 | |

*Significant at 0.05 level

**Girls performed significantly better than boys.**



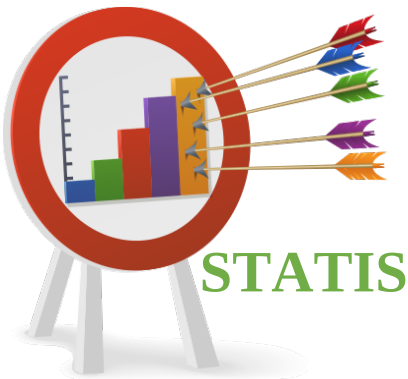
The table shows comparison between the performances of boys and girls in a particular subject. The t-value of 11.91 is significant at 0.05 level (mentioned below the table) with a mean of girls as 66.1 and that of boys as 60.1. It means that there is significant difference between the performance of boys and girls in the subject. The probability of this difference occurring by chance is less than 5%. The mean value of girls is higher than that of boys. It can be interpreted that girls are performing significantly better than boys.

The results have also been depicted pictographically. In this case, the girls performed significantly better than the boys. This is depicted by the girl with a medal.

## Qualitative Analysis

The purpose of the qualitative analysis is to provide insights into how candidates have performed in individual questions set in the question paper. This section is based on inputs provided by examiners from examination centres across the country. It comprises of question wise feedback on the performance of candidates in the form of *Comments of Examiners* on the common errors made by candidates along with *Suggestions for Teachers* to rectify/ reduce these errors. The *Marking Scheme* for each question has also been provided to help teachers understand the criteria used for marking. Topics in the question paper that were generally found to be difficult or confusing by candidates, have also been listed down, along with general suggestions for candidates on how to prepare for the examination/ perform better in the examination.

# QUANTITATIVE ANALYSIS



## STATISTICS AT A GLANCE

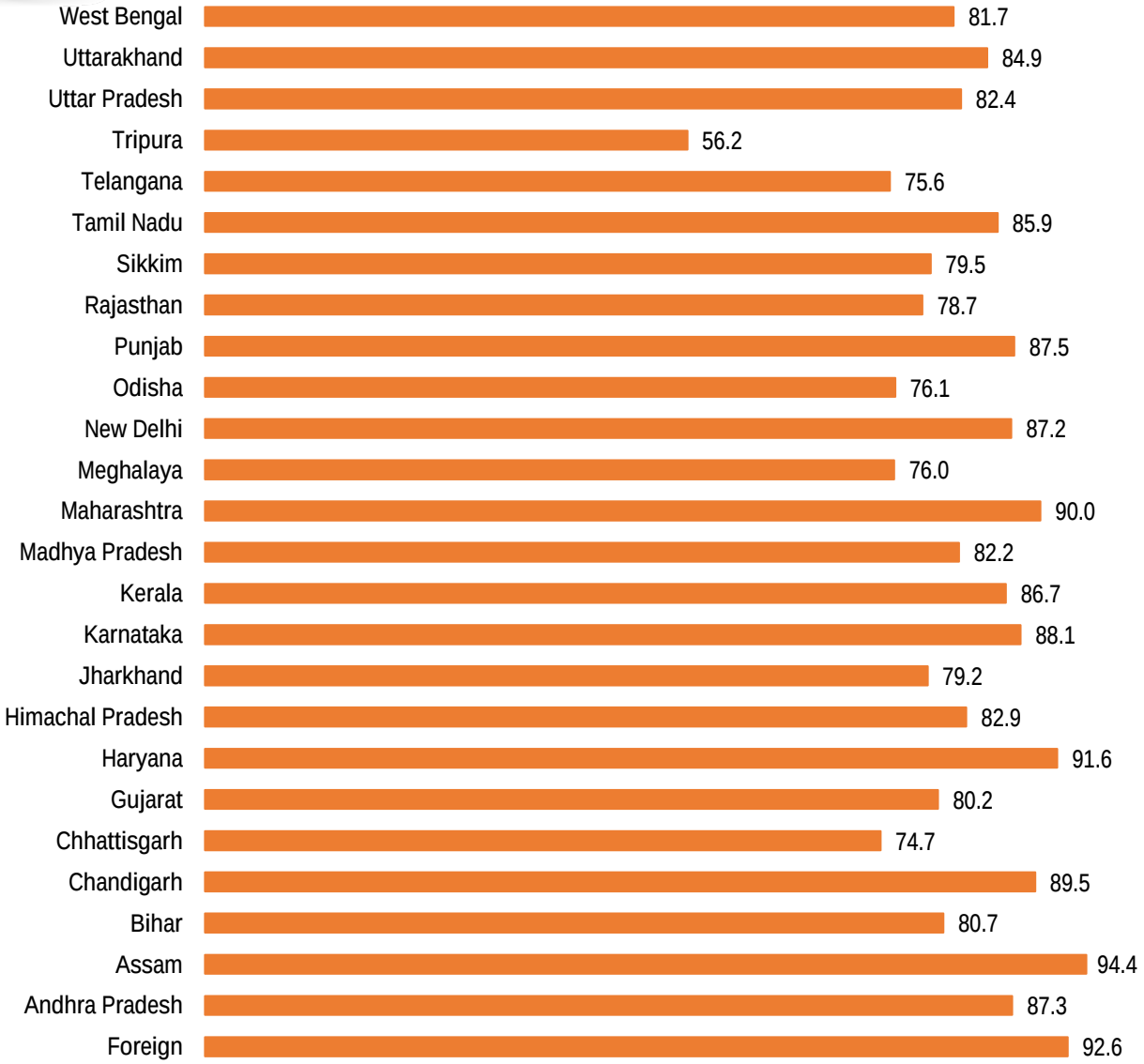
Total Number of Candidates: 21,960

Mean Marks: 82.5

Highest Marks: 100

Lowest Marks: 02

# PERFORMANCE (STATE-WISE & FOREIGN)




**The States of Assam, Haryana and Maharashtra secured highest mean marks. Mean marks secured by candidates studying in schools abroad were 92.6.**

# GENDER-WISE COMPARISON

**GIRLS**

Mean Marks: 82.5

Number of Candidates: 7,513

**BOYS**

Mean Marks: 82.5

Number of Candidates: 14,447

## Comparison on the basis of Gender

| Gender | N | Mean | SE | t-value |
|---|---|---|---|---|
| Girls | 7,513 | 82.5 | 0.18 | 0.18 |
| Boys | 14,447 | 82.5 | 0.14 | |

**No significant difference was observed between the performance of girls and boys.**

# REGION-WISE COMPARISON

**East**

**Mean Marks: 80.6**

**Number of Candidates: 7,105**

**Highest Marks: 100**
**Lowest Marks: 02**

**North**

**Mean Marks: 82.8**

**Number of Candidates: 12,114**

**Highest Marks: 100**
**Lowest Marks: 02**

**REGION**

**Mean Marks: 86.0**

**Number of Candidates: 1,772**

**Highest Marks: 100**
**Lowest Marks: 26**

**South**

**Mean Marks: 86.5**

**Number of Candidates: 912**

**Highest Marks: 100**
**Lowest Marks: 21**

**West**

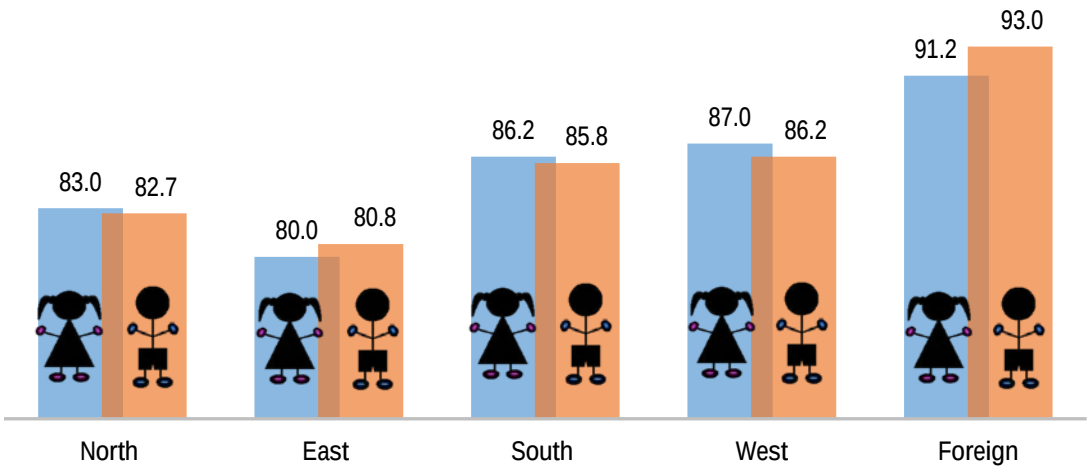
**Mean Marks: 92.6**

**Number of Candidates: 57**

**Highest Marks: 100**
**Lowest Marks: 63**

**Foreign**

## Mean Marks obtained by Boys and Girls-Region wise




### Comparison on the basis of Gender within Region

| Region | Gender | N | Mean | SE | t-value |
|---|---|---|---|---|---|
| North (N) | Girls | 4,393 | 83.0 | 0.23 | 1.00 |
| | Boys | 7,721 | 82.7 | 0.18 | |
| East (E) | Girls | 2,222 | 80.0 | 0.36 | -1.76 |
| | Boys | 4,883 | 80.8 | 0.25 | |
| South (S) | Girls | 589 | 86.2 | 0.55 | 0.62 |
| | Boys | 1,183 | 85.8 | 0.40 | |
| West (W) | Girls | 297 | 87.0 | 0.81 | 0.74 |
| | Boys | 615 | 86.2 | 0.62 | |
| Foreign (F) | Girls | 12 | 91.2 | 3.01 | -0.57 |
| | Boys | 45 | 93.0 | 1.27 | |

**In all the regions, no significant difference was observed between the performance of girls and boys.**

# MARK RANGES : COMPARISON GENDER-WISE

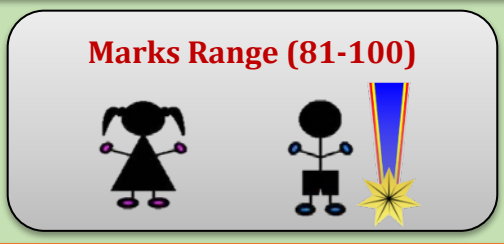
**Comparison on the basis of gender in top and bottom mark ranges**

| Marks Range | Gender | N | Mean | SE | t-value |
|---|---|---|---|---|---|
| Top Range (81-100) | Girls | 4,800 | 92.6 | 0.08 | -2.47* |
| | Boys | 9,376 | 92.8 | 0.06 | |
| Bottom Range (0-20) | Girls | 3 | 18.3 | 0.88 | 3.55* |
| | Boys | 9 | 9.6 | 2.31 | |

*Significant at 0.05 level

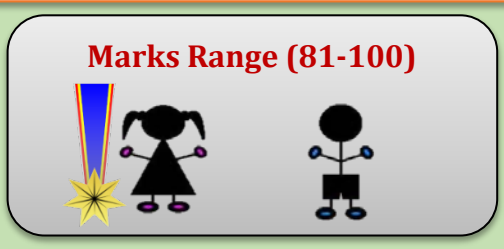
**Marks Range (81-100)**

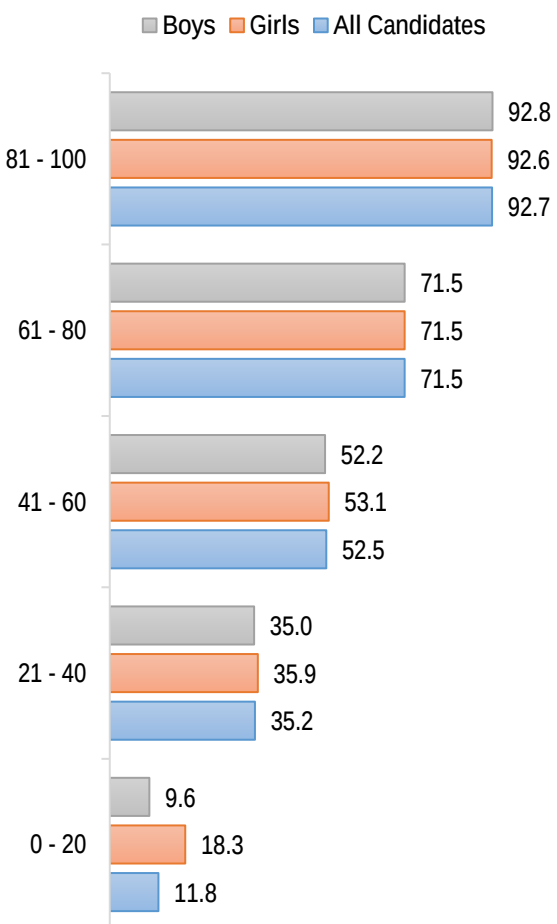
**Performance of boys was significantly better than the performance of girls .**




**Marks Range (0-20)**

**Performance of girls was significantly better than the performance of boys .**



# GRADES AWARDED : COMPARISON GENDER-WISE

**Comparison on the basis of gender in Grade 1 and Grade 9**

| Grades | Gender | N | Mean | SE | t-value |
|---|---|---|---|---|---|
| Grade 1 | Girls | 3,342 | 95.7 | 0.05 | -0.27 |
| | Boys | 6,733 | 95.7 | 0.04 | |
| Grade 9 | Girls | 30 | 30.4 | 0.88 | 1.97* |
| | Boys | 113 | 28.2 | 0.63 | |

*Significant at 0.05 level

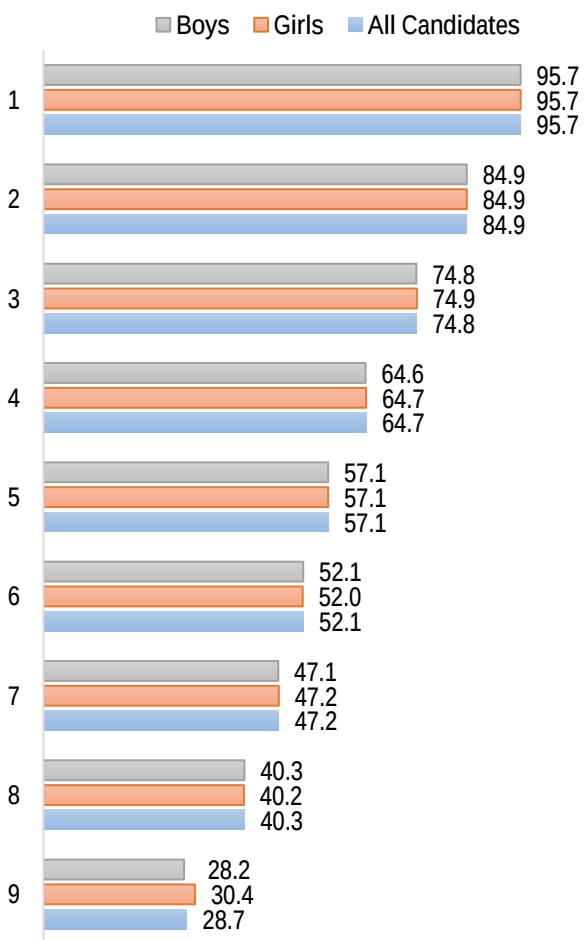
**Grade 1**

**No significant difference was observed between the performance of girls and boys.**

**Grade 9**

**Grade 9**

**Performance of girls was significantly better than the performance of boys.**



# PART I (20 Marks)

*Answer **all** questions.*

## Question 1

(a) Name and draw the logic gate represented by the following truth table, where A and B are inputs and X is the output. **[1]**

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(b) Write the canonical POS expression of: $F(P, Q) = \Pi ( 0,2 )$ **[1]**

(c) Find the dual of: $X.Y + X.Y' = X + 0$ **[1]**

(d) If $F (A, B, C) = A'.B'.C' + A'.B.C'$ then find $F'$ using De Morgan's Law. **[1]**

(e) If A= "It is cloudy" and B= "It is raining", then write the proposition for: **[1]**

(i) Contrapositive

(ii) Converse

## Comments of Examiners

(a) Most of the candidates answered this question quite well. However, some candidates only drew the logic gate without mentioning the name. A few candidates drew the logic circuit for (A'B + AB') instead of drawing it for the logic gate represented by the given truth table. A few others mentioned XNOR instead of XOR.

(b) Majority of the candidates wrote the correct answer. However, a few common errors made by the candidates were:

(i) Wrote the direct answer without showing the working.

(ii) Wrote the SOP expression instead of the required POS expression.

(iii) Used truth tables to find the canonical POS expression.

### ***Suggestions for teachers***

- Ensure that students practice all the logic gates along with their truth tables and symbols.
- Give adequate practice to the students in the conversion of SOP and POS and *vice-versa*.
- Explain both Canonical and Cardinal forms to the students.
- Give more practice on principle of duality to show that there exists another equation in every Boolean equation. Explain the difference between complement and duality with examples.

(iv) Interchanged maxterms with minterms and *vice versa.*

(c) A number of candidates answered it correctly. However, the common errors made by some candidates were:
(i) changed complements but not interchanged 0's and 1's.
(ii) wrote the definition.
(iii) found the dual of LHS and RHS separately instead of as one single equation.
(iv) did not put the brackets.

(d) Most candidates answered this question quite well. The common errors found in some answer scripts were:
(i) expressed the answer directly without showing the working/steps.
(ii) changed the operators improperly.
(iii) simplified the equation and applied De Morgan's law to the reduced expression.
(iv) stated the De Morgan's law, but not applied it to the given expression.
(v) wrote the dual instead of complementation.

- Ensure that students are given adequate practice on complementation using De Morgan's law knowing its usefulness in reducing / minimizing a Boolean expression.
- Give adequate practice to the students on the use of all the laws of Boolean algebra and Propositional logic.
- Teach with examples the use of terms like Converse, Inverse and Contrapositive etc.
- Explain the use of the symbols ∧,∨, ~, => and <=> in a proposition.
- Encourage students to show working/steps.

(e) Most of the candidates answered this question correctly. Some candidates got confused with the proposition and swapped the answer of Contrapositive with Converse and *vice versa*. A few candidates used the symbolic representation while a few others represented it in the statement form. Variety of answers were given as candidates changed the proposition *not cloudy* as *clear* and *not raining* as *dry*.

## MARKING SCHEME

### Question 1

| | |
|---|---|
| (a) | Name of the Gate: XOR<br>Symbol of the gate:<br>  |
| (b) | F(P,Q) = Π ( 0, 2 )<br>= ( 00, 10 )<br>= ( P + Q) . (P′ + Q ) |
| (c) | Find the dual of : X.Y + X.Y′ = X + 0<br>Dual = (X + Y ) . (X + Y′ ) = X. 1 |
| (d) | F(A,B,C) = A'B'C' + A'BC'<br>F'= ( A'B'C' + A'BC' )'<br>= ( A'B'C' )' . ( A'BC')'<br>= ( A + B + C ) . ( A + B' + C) |
| (e) | (i) If it is not raining, then it is not cloudy ( ~B => ~A)<br>(ii) If it is raining, then it is cloudy ( B => A) |

# Question 2

(a) What is an *Interface*? How is it different from a *class*? **[2]**

(b) A matrix ARR[ – 4…6, 3….8] is stored in the memory with each element requiring 4 bytes of storage. If the base address is 1430, find the address of ARR[3][6] when the matrix is stored in **Row Major Wise**. **[2]**

(c) Convert the following *infix notation* to *postfix* form: **[2]**

(A + B * C ) – ( E * F / H ) + J

(d) Compare the two complexities $O(n^2)$ and $O(2^n)$ and state which is better and why. **[2]**

(e) State the difference between *internal nodes* and *external nodes* of a binary tree structure. **[2]**

## Comments of Examiners

(a) Most of the candidates answered this question quite well, especially the difference between an interface and a class. Some candidates wrote a vague definition of Interface while some others wrote the definition of User Interface of Operating System. A few candidates wrote only the examples without the definition while a few others wrote the definition without examples.

(b) Majority of the candidates were able to answer this question well. However, some candidates wrote the answer directly without showing the working/formula. Some candidates made errors in their calculations including the calculations involving the size of the matrix. A few candidates solved the question by Column major instead of Row major.

(c) Many candidates were able to solve this problem correctly. Some candidates wrote the correct answer but did not show the working. Some candidates applied the postfix correctly but could not derive the final answer due to wrong operator precedence. A few candidates followed BODMAS while a few others did not follow the concept of left-to-right.

(d) While most candidates attempted this question well, some candidates gave only examples to illustrate the answers. Some drew graphs to illustrate the complexities of $O(n^2)$ and $O(2^n)$. However, a few candidates did not mention which is better of the two while a few others did not attempt this part of the question.

(e) Majority of the candidates were able to answer this question quite well giving definitions / examples. However, some candidates wrote vague answers

### ***Suggestions for teachers***

- Revise and integrate the concepts studied by the students in the previous class syllabus with the present class and ensure that the theory of each chapter/topic is covered.
- Elaborate basic concepts of Java along with programs.
- Emphasise to the students to learn the definitions precisely and supported by examples if possible.
- Give more practice to the students to calculate addresses using Row major and Column major wise and in solving equations.
- Explain the different terms used in address calculations.
- Make the students solve ample number of problems involving conversion of Infix to Postfix notation, the order of precedence and also the Polish Stack method.
- Clarify to the students, with examples, all the three cases of complexities with their factor that influence.
- Explain the various programs in which complexity differs, for example- loops, nested loops, conditional, recursion etc.
- Give the explanation of root, height, depth, size, degree, siblings, nodes (internal and external), levels, tree traversals etc. by using a binary tree diagram and ensure that adequate practice to solve questions on this topic is given to the students.

not related to the topic. Some drew the binary tree and indicated the internal and external nodes in it. A few candidates swapped the definition of internal nodes and external nodes.

## MARKING SCHEME

## Question 2

| | |
|---|---|
| (a) | **Interface**: An interface is a blueprint of a class having static constants and abstract methods An interface is a reference type in Java.<br>A class that contains public static final data types and declaration of abstract method only.<br><br>Class contains static or non-static data members with methods defined. Interface support multiple inheritance, where as a class does not. Interfaces are implemented by classes whereas sub classes are extended from super class. |
| (b) | **Row Major Wise**: ARR[i][j] = BA + W [ (i – $l_r$ )* col + (j – $l_c$]<br>Putting the values: = 1430 + 4[ (3 –(-4)) * 6 + (6-3)]<br>= 1430 + 4[ 42 + 3]<br>= 1430 + 180<br>= **1610** |
| (c) | Infix to Postfix: ( A + B * C ) – ( E * F / H ) + J<br>= ( A + BC* ) – ( EF* / H ) + J<br>= ( ABC*+) – (EF*H/) +J<br>= ( ABC*+EF*H/-) + J<br>= **ABC*+EF*H/-J+** |
| (d) | O ($n^2$) : Complexity for nested loop with equal number of statements.<br>O($2^n$): Complexity for recursive algorithms that solve a problem of n steps.<br>O($n^2$) is better than O ($2^n$) because if the size of n=5, then O($n^2$) would require 25 executions, whereas O($2^n$) would require 32 executions. |
| (e) | Internal nodes: Nodes which have sub nodes or children.<br>External nodes: Nodes which do not have any sub nodes. They are also known as leaves. |

# Question 3

The following function **Mystery( )** is a part of some class. What will the function Mystery ( ) return when the value of **num=43629**, **x=3** and **y=4** respectively? Show the dry run/ working. **[5]**

```
int  Mystery( int num, int x, int y)
 {
      if(num<10)
          return num;
      else
        {
            int z = num % 10;
            if( z % 2 = = 0 )
                return z*x + Mystery( num/10,x,y);
```

```
        else
            return z*y + Mystery( num/10,x,y);
    }

}
```

## Comments of Examiners

Many candidates answered this question quite well. However, some candidates were unable to accumulate the result along with the recursive calling. Some candidates were confused with the operators (% and /) and with the odd and even checking / processing. In some answer scripts the intermediate result was returned with each recursive call. A few candidates either gave the output directly without showing the working / steps or showed vague working / output.

### *Suggestions for teachers*

- Give adequate practice to the students on Standard programs, especially Recursive programs and in programs using conditions / looping and other output related programs.
- Show the dry run / working of programs and emphasise that working is necessary.

## MARKING SCHEME

## Question 3

Mystery ( 43629 , 3 , 4 )

9 * 4 + Mystery (4362, 3, 4)

2 * 3 + Mystery ( 436, 3, 4)

6 * 3 + Mystery ( 43, 3, 4)

3 * 4 + Mystery (4, 3, 4)

4

Final output: 4 + 12 + 18 + 6 + 36 = **76**

# PART – II (50 Marks)

*Answer **six** questions in this part, choosing **two** questions from Section A, **two** from Section B and **two** from Section C.*

# SECTION - A

*Answer **any two** questions.*

## Question 4

(a) Given the Boolean function **F(A, B, C, D) = Σ ( 0, 2, 3, 4, 5, 8, 10, 11, 12, 13 ).**

(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). **[4]**

(ii) Draw the logic gate diagram for the reduced expression using only NAND gates. Assume that the variables and their complements are available as inputs. **[1]**

(b) Given the Boolean function: **F(P, Q, R, S) = π ( 0, 1, 2, 8, 9, 11, 13, 15 ).**

(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). **[4]**

(ii) Draw the logic gate diagram for the reduced expression using only NOR gates. Assume that the variables and their complements are available as inputs **[1]**

## Comments of Examiners

(a)(i) Most candidates fared well in this part. However, there were several candidates who were not clear in the *Map rolling* concept as they were either unable to draw correctly the K-Map for the SOP expression or used different variables to draw the K-Map instead of that given in the question. Some candidates drew the POS K-Map instead of SOP.A few candidates included redundant groups in the final expression which was not required.

(ii) Most of the candidates answered this part correctly. Some candidates drew the logic circuit using basic gates while some others drew vague diagrams with different shapes instead of NAND gates. A few candidates did not show the labelling and flow lines.

(b) (i) Many candidates answered this sub part of the question correctly. Some candidates made an error either in place value or while putting variables in K-Map. Some candidates reduced the groups by laws while some included the redundant group in the final expression. A few candidates drew the K-Map incorrectly while a few others drew the SOP K-Map instead of POS.

### ***Suggestions for teachers***

- Make the students reduce SOP and POS expressions using K-Map simultaneously.
- Instruct the students not to include the redundant group in the final expression.
- Give adequate practice in drawing the K-Map, filling the K-Map with 0's and 1's, marking the groups and reducing the groups.
- Ensure that intensive practice is given to the students in drawing logic circuits using universal gate / basic gates.
- Emphasise on arranging the variables in proper order and the importance of cell values corresponding with the variables.
- Explain clearly how the groups are framed and reduced and that redundant groups are not to be included in the final reduced expression.

(ii) Majority of the candidates answered correctly. However, there were some candidates who drew the logic circuit using basic gates while some others drew vague diagrams with different shapes instead of NOR gates. A few candidates did not show the labelling and flow lines.

## MARKING SCHEME

## Question 4

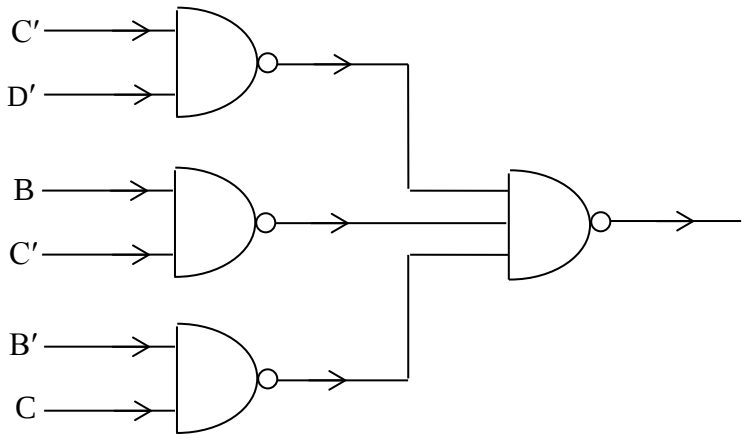
(a) F(A,B,C,D) = $\sum$ (**0, 2 , 3, 4, 5, 8, 10, 11, 12, 13** )

| | **C'D'** | **C'D** | **CD** | **CD'** |
|---|---|---|---|---|
| **A'B'** | 0 **1** | 1 0 | 3 **1** | 2 **1** |
| **A'B** | 4 **1** | 5 **1** | 7 0 | 6 0 |
| **AB** | 12 **1** | 13 **1** | 15 0 | 14 0 |
| **AB'** | 8 **1** | 9 0 | 11 **1** | 10 **1** |

There are three quads:

Quad 1 ($m_0 + m_4 + m_{12} + m_8$) = **C'D'** Quad 2 ($m_4 m_5 + m_{12} + m_{13}$) = **BC'**
Quad 3 ($m_2 + m_3 + m_{10} + m_{11}$) = **B'C** Quad 4 ($m_0 + m_2 + m_8 + m_{10}$) = **B'D'**

Hence, F (A, B, C, D) = **C'D' + BC' + B'C** OR F (A, B, C, D) = **B'D' + BC' + B'C**



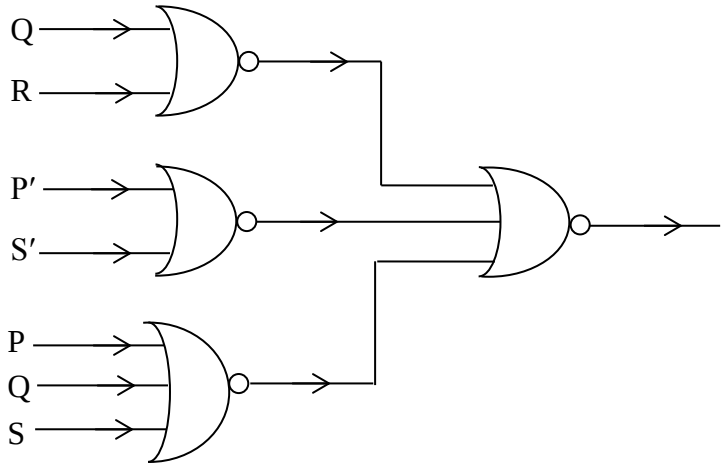
(b) F (P, Q, R, S) = π **(0, 1, 2, 8, 9, 11, 13, 15 )**

| | R+S | R+S' | R'+S' | R'+S |
|---|---|---|---|---|
| P+Q | 0 **0** | 1 **0** | 3 1 | 2 **0** |
| P+Q' | 4 1 | 5 1 | 7 1 | 6 1 |
| P'+Q' | 12 1 | 13 **0** | 15 **0** | 14 1 |
| P'+Q | 8 **0** | 9 **0** | 11 **0** | 10 1 |

There are two quads and one pair:

Quad 1: ($M_0M_1M_8M_9$) = **Q + R** Quad 2: ($M_9M_{11}M_{13}M_{15}$) = **P' + S'**
Pair: ($M_0M_2$) = **P + Q +S**

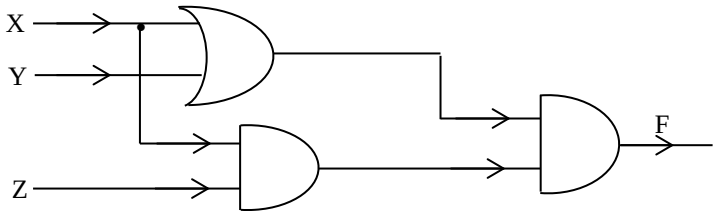
Hence, F (P, Q, R, S) = **(Q + R). (P' + S'). (P + Q + S)**




# Question 5

(a) How is a *decoder* different from a *multiplexer*? Write the truth table and draw the logic circuit diagram for a 3 to 8 decoder and explain its working. **[5]**

(b) From the logic circuit diagram given below, derive the Boolean expression and simplify it to show that it represents a logic gate. Name and draw the logic gate. **[3]**



(c) Using a truth table, state whether the following proposition is a Tautology, Contradiction or Contingency: **[2]**

$\sim(P \Rightarrow Q) \Leftrightarrow (\sim P \vee Q)$

## Comments of Examiners

(a) Most of the candidates answered this question quite well, especially the logic diagram. However. a few common errors made by some candidates were:

- Wrote the definition of encoder instead of decoder.
- Drew the OR gate instead of AND gate.
- Used the OR gate to combine parallel lines to a single serial line which was not required.
- Drew the decoder only for the numbers 3 to 8 and not octal decoder.
- Did not show the working / explanation.

(b) The first two outputs, i.e. intermediate parts were answered correctly by majority of the candidates, but the final expression and its reducing was confusing to some candidates. Some candidates were not clear about the symbol of the final gate for the reduced expression. A few candidates drew the gate correctly but overlooked the naming of the gate and *vice versa*.

(c) Well answered by most of the candidates. However, some candidates were confused with the symbols~, ∧, ∨ and =>. Some candidates used the truth table for 3variables instead of for 2 variables while a few others used the law instead of the Truth Table. In some answer scripts, one column (=>) was incorrect which resulted in an incorrect result. Some candidates did not mention the proposition being a contradiction.

### *Suggestions for teachers*

- Give adequate practice to the students so that they become conversant with the circuit diagram, truth table, expression, definition, and use for all applications of Boolean algebra i.e. Half adder, Full adder, Encoders, Decoders etc.
- Explain to the students, the purpose and working of the gates for the said applications.
- Make the students practice the derivation of expression from a logic diagram and reducing it.
- Explain in detail the basic gates i.e. AND, OR and NOT gate.
- Clarify to the students the laws of Boolean algebra and Propositional logic.
- Lay stress on proving of all the laws.
- Explain the use of the symbols Λ,V, ~, => and <=> in a proposition.
- Give adequate practice in the concept of tautology, contingency or a contradiction and Truth tables for conditional (=>) and bi-conditional (<=>).
- Encourage students to read the question carefully and answer accordingly.

## MARKING SCHEME

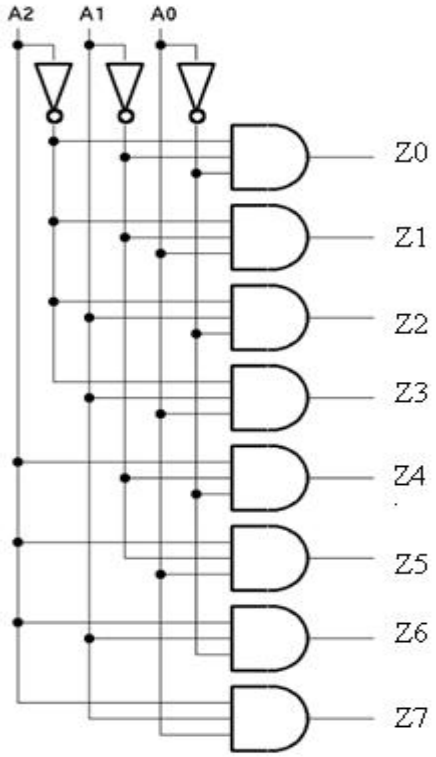
## Question 5

(a) A decoder is a combinational circuit which inputs n lines and outputs $2^n$ or fewer lines where as a multiplexer is a combinational circuit which inputs parallel data (from decoders) and outputs one serial data using OR gate.

Truth table for a 3 to 8 decoder:

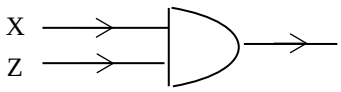
| INPUT | | | OUTPUT |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

Circuit diagram of a 3 to 8 decoder:




Working: It converts Low Level to High Level (Binary to Octal, Decimal or Hexadecimal).
Example: If input is $(010)_2$ i.e. A'BC' the output will be 2, i.e. decimal value of 010 and so on.

(b) Expression : F = (X+Y). XZ
= XZ +XYZ
= XZ
It represents AND gate.




(c) **~(P =>Q) <=> (~P ∨ Q) is a Tautology, Contradiction or Contingency.**

| P | Q | ~P | (P=>Q) | ~(P=>Q) | (~P∨Q ) | ~(P=>Q)<=>(~P∨Q ) |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |

Hence, it is a **Contradiction**

# Question 6

(a) The owner of a company pays bonus to his salesmen as per the criteria given below: **[5]**

- If the salesman works overtime for more than 4 hours but does not work on off days/holidays.

**OR**

- If the salesman works when festival sales are on and updates showroom arrangements.

**OR**

- If the salesman works on an off day/holiday when the festival sales are on.

The inputs are:

| **INPUTS** | |
|---|---|
| O | Works overtime for more than 4 hours |
| F | Festival sales are on |
| H | Working on an off day/holiday |
| U | Updates showroom arrangements |

(In all the above cases 1 indicates yes and 0 indicates no.)

Output : **X** [1 indicates yes, 0 indicates no for all cases]

Draw the truth table for the inputs and outputs given above and write the **POS** expression for **X(O,F,H,U)**.

(b) What is a *half adder*? Write the truth table and derive an SOP expression for *sum* and *carry* for a half adder. **[3]**

(c) Simplify the following expression, using Boolean laws: **[2]**

$$( X + Z ) . ( X.Y + Y.Z' ) + X.Z + Y$$

## Comments of Examiners

(a) Majority of the candidates answered this question very well. However, some candidates did not mention the final expression. Some candidates were confused with the SOP/POS expression and interchanged 0's and 1's and vice-versa. Some candidates reduced the expression using K-Map which was not required. Decoding the criteria was not clear to some of the candidates. A few candidates drew logic diagrams which were not required.

(b) Most of the candidates were able to answer this question. However, some candidates wrote a very vague definition of half adder with the terms 'adds' and 'binary digits' missing in the definition. Some candidates drew the truth table and expression for full adder which was not required. Some candidates drew two different diagrams for sum

### *Suggestions for teachers*

- Make the students practice Truth Table with 4 input variables (i.e. 16 combinations).
- Explain to the students the Propositional logic to find the criteria for the output.
- Train students to express the final expression written either in Canonical form or Cardinal form for both SOP and POS expressions.
- Give adequate practice to the students to make them conversant with the circuit diagram, truth table, expression, definition, and applications of Boolean algebra.
- Encourage the students to mention laws along with the working.

and carry while a few others drew the block diagram instead of the logic circuit. Some candidates interchanged the sum and carry columns.

- Make the students practice reducing / simplifying the expressions covering maximum laws.

(c) Most candidates attempted this question correctly. However, some candidates wrote only the final answer without showing the working/steps or mentioning the laws. Some candidates used a long method, there by wasting their time. A few candidates applied the incorrect laws.

## MARKING SCHEME

## Question 6

(a)

| O | F | H | U | X (OUTPUT) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | 1 | **0** |
| 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 1 | 1 | **0** |
| 0 | 1 | 0 | 0 | **0** |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | **0** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**SOP Expression: X (O ,F, H, U) = Π (0, 1, 2, 3, 4, 10, 11) OR**
**X = (O+F+H+U).(O+F+H+U').(O+F+H'+U).(O+F+H'+U').(O+F'+H+U)**
**(O'+F+H'+U).(O'+F+H'+U')**

(b) Half adder is a combinational circuit which adds two input binary digits and outputs two binary digits (Sum and Carry)

| A | B | Sum | CARRY |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

SOP Expression: Sum = A'B + AB' **OR** Sum = A⊕ B
Carry = A.B

(c) Simplify: ( X + Z ) . (X Y + Y Z') + X Z + Y
= XY + XYZ' + XYZ + XZ + Y
= XY (1+Z' +Z) +XZ + Y
= XY + Y + XZ
= Y + XZ

# SECTION – B

*Answer **any two** questions.*

## Question 7

Design a class **ArmNum** to check if a given number is an Armstrong number or not. **[10]**

[A number is said to be Armstrong if sum of its digits raised to the power of length of the number is equal to the number]

Example : 371 $= 3^3 + 7^3 + 1^3$
1634 $= 1^4 + 6^4 + 3^4 + 4^4$
54748 $= 5^5 + 4^5 + 7^5 + 4^5 + 8^5$

Thus 371, 1634 and 54748 are all examples of Armstrong numbers.

Some of the members of the class are given below:

| | | |
|---|---|---|
| **Class name** | : | **ArmNum** |
| **Data members/instance variables:** | | |
| n | : | to store the number |
| l | : | to store the length of the number |
| **Methods/Member functions:** | | |
| ArmNum (int nn) | : | parameterized constructor to initialize the data member n=nn |
| int sum_pow(int i) | : | returns the sum of each digit raised to the power of the length of the number using **recursive technique**<br>eg. 34 will return $3^2 + 4^2$ (as the length of the number is 2) |
| void isArmstrong( ) | : | checks whether the given number is an Armstrong number by invoking the function *sum_pow( )* and displays the result with an appropriate message |

Specify the class **ArmNum** giving details of the **constructor( )**, **int sum_pow(int)** and **void isArmstrong( ).** Define a **main( )** function to create an object and call the functions accordingly to enable the task.

## Comments of Examiners

Most candidates answered this question quite well. However, in some answer scripts following lapses were observed:

- used additional instance variables which was not in accordance with the question.
- unclarity of the concept of recursion.
- the parameters in the method sum pow() was not properly understood.
- recursive case was improperly done.
- the base case was missing in some of the answer scripts.
- attempted the recursive method without using the recursive technique and used iteration (loops) instead.
- problem in the isarmstrong() method as some candidates did not invoke the sum pow() method properly.
- did not calculate the length of the number.
- object creation and method calling was not done properly in the main() function. some did not write the main() method.
- documentation/comments were missing.

### *Suggestions for teachers*

- Explain thoroughly to the students of recursion and its techniques with examples.
- Give proper knowledge of base case and recursive case to the students for every program using recursive technique.
- Train students to Invoke function within another function.
- Explain to the students the difference between iteration and recursion.
- Teach students thoroughly the instance variables and their accessibility.
- Advise the students to read the questions carefully and answer according to their requirements.

## MARKING SCHEME

## Question 7

```
import java.util.*;
public class ArmNum
{    int n,l;
     static Scanner x=new Scanner(System.in);
     ArmNum(int nn)
    { n=nn;
      l=Integer.toString(n).length();
     }
     int sum_pow(int i)
     { if(i==0)
           return 0;
       else
           return (int)Math.pow(i%10,l) + sum_pow(i/10);
      }
     void isArmstrong()
     {  if(sum_pow(n)==n)
            System.out.println(n + "is an Armstrong number");
        else
            System.out.println(n + "is not an Armstrong number");
      }
      static void main()
     { ArmNum obj=new ArmNum(546748);
       obj.isArmstrong();
      }
}
```

# Question 8

Design a class **MatRev** to reverse each element of a matrix. **[10]**

Example:

| | | |
|---|---|---|
| 72 | 371 | 5 |
| 12 | 6 | 426 |
| 5 | 123 | 94 |

becomes

| | | |
|---|---|---|
| 27 | 173 | 5 |
| 21 | 6 | 624 |
| 5 | 321 | 49 |

Some of the members of the class are given below:

| | | |
|---|---|---|
| **Class name** | : | **MatRev** |
| **Data members/instance variables:** | | |
| arr[ ][ ] | : | to store integer elements |
| m | : | to store the number of rows |
| n | : | to store the number of columns |
| **Member functions/methods:** | | |
| MatRev(int mm, int nn) | : | parameterised constructor to initialise the data members m = mm and n = nn |
| void fillarray( ) | : | to enter elements in the array |
| int reverse(int x) | : | returns the reverse of the number x |
| void revMat( MatRev P) | : | reverses each element of the array of the parameterized object and stores it in the array of the current object |
| void show( ) | : | displays the array elements in matrix form |

Define the class **MatRev** giving details of the **constructor( ), void fillarray( ), int reverse(int)**, **void revMat(MatRev)** and **void show( )**. Define the **main( )** function to create objects and call the functions accordingly to enable the task.

## Comments of Examiners

Majority of the candidates answered this question well. Constructor was properly defined by most of the candidates. However, common errors made by some candidates were:

- not declared memory to the array using the *new* operator.
- *passing of objects* was done improperly.
- created a different array instead of passing objects.
- object was not created properly in the main() method.
- created three objects instead of two in the main( ).
- In a few answer scripts the calling of the method revMat() was incorrect. The returning type of the method reverse() was confusing to a few students.

## *Suggestions for teachers*

- Train students in *passing of objects to a function through parameters.*
- Explain to the students the working on one-dimensional and two-dimensional arrays and pass by value and pass by reference in detail with various examples.
- Teach students the dynamic binding concepts and dot (.) operator in detail.
- Explain the use of flag variables.
- Advise the students to solve a question as per its rubric.

## MARKING SCHEME

## Question 8

```
import java.util.*;
public class MatRev
{
   int arr[ ][ ],m,n;
   static Scanner x=new Scanner(System.in);
   MatRev(int nn,int mm)
   { m=mm;
     n=nn;
     arr=new int[m][n];
   }
   void fillarray()
   { for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
           arr[i][j]=x.nextInt();
    }
   int reverse(int x)
   { int s=0,c=0;
     while(x!=0)
     {  c=x%10;
        s=s*10 + c;
        x=x/10;
      }
     return s;
   }
  void revMat(MatRev P)
  { for(int i=0;i<m;i++)
       for(int j=0;j<n;j++)
          arr[i][j]=reverse(P.arr[i][j]);
   }
  void show()
   { for(int i=0;i<m;i++)
      {   System.out.println();
          for(int j=0;j<m;j++)
             System.out.print(arr[i][j] + "\t");
       }
    }
   static void main()
   { MatRev obj = new MatRev(3,4);
     MatRev obj1=new MatRev(3,4);
     obj.fillarray();
     System.out.print("\n Original Array");
     obj.show();
     obj1.revMat(obj);
     System.out.print("\n Reverse array");
     obj1.show();
   }
}
```

# Question 9

A class **Rearrange** has been defined to modify a word by bringing all the vowels in the word at the beginning followed by the consonants. **[10]**

Example: ORIGINAL becomes OIIARGNL

Some of the members of the class are given below:

| | | |
|---|---|---|
| **Class name** | : | **Rearrange** |
| **Data member/instance variable:** | | |
| wrd | : | to store a word |
| newwrd | : | to store the rearranged word |
| **Member functions/methods:** | | |
| Rearrange( ) | : | default constructor |
| void readword( ) | : | to accept the word in UPPER case |
| void freq_vow_con( ) | : | finds the frequency of vowels and consonants in the word and displays them with an appropriate message |
| void arrange( ) | : | rearranges the word by bringing the vowels at the beginning followed by consonants |
| void display( ) | : | displays the original word along with the rearranged word |

Specify the class **Rearrange**, giving the details of the **constructor( )**, **void readword( )**, **void freq_vow_con( ), void arrange( )** and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

## Comments of Examiners

Most of the candidates answered this question well, especially the arrange( ) method, the object creation and the method calling in the main ( ). However, some candidates accepted the word and converted it into upper case. Different methods/logic were used by some candidates to count the frequency of vowels and consonants. In a few answer scripts the ASCII values were incorrect.

### *Suggestions for teachers*

- Give adequate practice to the students to extract characters from words, words from sentences and sentences from paragraphs.
- Give wider exposure to the students to string manipulation related programs by adopting different methods /logic.
- Advise students to practice extracting and checking each character for vowel, consonant or digit.
- Familiarise students with the concept of constructors to initialize a string and other data members and give adequate practise to deal with the questions based on the concept.

## MARKING SCHEME

## Question 9

```
import java.util.*;
public class Rearrange
{
    String wrd,newwrd;
    static Scanner x=new Scanner(System.in);
    Rearrange(){}
    void readword()
    { System.out.println("Enter a word" );
      wrd=x.next();
      wrd=wrd.toUpperCase();
     }
    void freq_vow_con()
    { int s=0,s1=0;
      char ch;
      for(int i=0;i<wrd.length();i++)
      { ch=wrd.charAt(i);
           if("AEIOU".indexOf(ch)!=-1)
             s++;
       }
       s1= wrd.length()-s;
       System.out.println("vowels = "+ s);
       System.out.println("consonants = " + s1);
     }
   void arrange()
   {   char ch;
       String p="",q="";
       for(int i=0;i<wrd.length();i++)
       {  ch=wrd.charAt(i);
          if("AEIOU".indexOf(ch)!=-1)
            p +=ch;
         else
           q +=ch;
       }
       newwrd= p+q;
    }
    void display()
    {  System.out.println("Original word = "+ wrd);
       System.out.println("Rearranged word = "+ newwrd);
     }
    static void main()
    {  Rearrange obj=new Rearrange();
       obj.readword();
       obj.freq_vow_con();
       obj.arrange();
       obj.display();
     }
}
```

# SECTION – C

*Answer **any two** questions.*

## Question 10

A super class **Record** contains names and marks of the students in two different single dimensional arrays. Define a sub class **Highest** to display the names of the students obtaining the highest mark. **[5]**

The details of the members of both the classes are given below:

| | | |
|---|---|---|
| **Class name** | : | **Record** |
| **Data member/instance variable:** | | |
| n[ ] | : | array to store names |
| m[ ] | : | array to store marks |
| size | : | to store the number of students |
| **Member functions/methods:** | | |
| Record(int cap) | : | parameterized constructor to initialize the data member size = cap |
| void readarray() | : | to enter elements in both the arrays |
| void display( ) | : | displays the array elements |
| **Class name:** | | **Highest** |
| **Data member/instance variable:** | | |
| ind | : | to store the index |
| **Member functions/methods:** | | |
| Highest(…) | : | parameterized constructor to initialize the data members of both the classes |
| void find( ) | : | finds the index of the student obtaining the highest mark and assign it to '**ind**' |
| void display( ) | : | displays the array elements along with the names and marks of the students who have obtained the highest mark |

*Assume that the super class **Record** has been defined*. Using the **concept of inheritance**, specify the class **Highest** giving the details of the **constructor(…)**,**void find( )** and **void display( )**.

**The super class, main function and algorithm need NOT be written**.

## Comments of Examiners

Majority of the candidates were unable to answer this question correctly. Some common observations/errors were:

- concept of inheritance and constructor with inheritance was not clear.
- the keywords extends and super were missing.
- accessing the members of the super class by the derived class was not clear.
- declared the base class which was not required.
- data members were not declared properly.
- sorted the array to find the highest mark.
- wrote the algorithm instead of a program.
- the readarray( ) function in the super class was not invoked in the derived class.

### *Suggestions for teachers*

- Explain clearly to the students the concept of overriding in inheritance with examples laying stress on the keywords 'extends' and 'super' in inheritance.
- Clarify the use of constructor of the base class.
- Explain to the students the different visibility modes and their accessing capability.
- Ensure that the students are well versed with the concept of calling the member function from the super class to the derived class.
- Instruct the students to read the question carefully and answer accordingly.

## MARKING SCHEME

## Question 10

```
import java.util.*;
class Highest extends Record
{   int ind;
    Highest(int cap)
    {   super(cap);
        ind=-1;
    }
   void find()
   {  readarray();
      int hm=m[0];
      for (int i=0;i<size;i++)
      {  if (m[i]>hm)
          {  hm=m[i];
             ind=i;
          } }  }
   void display()
   {  super.display();
      for(int i=0;i<size;i++)
      { if(m[i]==m[ind])
          System.out.println("Highest obtained by  "+n[i] + "  marks  "+m[i]);
       }
     }
 }
```

# Question 11

A linear data structure enables the user to add address from rear end and remove address from front. Define a class **Diary** with the following details:

| | | |
|---|---|---|
| **Class name** | **:** | **Diary** |
| **Data members / instance variables:** | | |
| Q[ ] | : | array to store the addresses |
| size | : | stores the maximum capacity of the array |
| start | : | to point the index of the front end |
| end | : | to point the index of the rear end |
| **Member functions:** | | |
| Diary (int max) | : | constructor to initialize the data member size=max, start=0 and end=0 |
| void pushadd(String n) | : | to add address in the diary from the rear end if possible, otherwise display the message " NO SPACE" |
| String popadd( ) | : | removes and returns the address from the front end of the diary if any, else returns "**?????**" |
| void show( ) | : | displays all the addresses in the diary |

(a) Specify the class **Diary** giving details of the functions **void pushadd(String)** and **String popadd( ).** Assume that the other functions have been defined. **[4]**

**The main function and algorithm need NOT be written**.

(b) Name the entity used in the above data structure arrangement. **[1]**

## Comments of Examiners

(a) Most of the candidates were unable to answer this question. The concept of queue was not clear to many candidates. The common errors made by the candidates were:

- the condition / logic for underflow and overflow was not answered correctly.
- increment / decrement of start and end index was not done properly.
- used the queue to store integers instead of string.

A few candidates found the methods pushadd() and popadd() to be difficult. Some candidates attempted this question using other method which was not required.

(b) Well answered by most candidates. Some candidates were unable to state the entity and wrote vague answers. Some candidates over looked this part while some have mentioned the principles of FIFO as entity.

### *Suggestions for teachers*

- Show the students as to how the stack or a queue performs, while explaining all its applications / uses.
- Give adequate practice to the students in data structure programs like the stacks, queues, de- queues etc.
- Emphasise implementation of stacks, queues and de queues using arrays.
- Explain the concept taking the base as an array.
- Explain the concept of LIFO and FIFO to the students with examples related to the real world.

## MARKING SCHEME

## Question 11

(a)
```
class Diary
{  String Q[ ];
    int size, start, end;
void pushadd(String n)
{  if(end<size-1)
      Q[++end]=n;
   else
      System.out.println(" NO SPACE");
}
String popadd()
{  if(start!=end)
       return Q[++start];
   else
       return "?????";
 } }
```

(b) The entity used in the above data structure is **Queue**.

# Question 12

(a) A linked list is formed from the objects of the class **Node**. The class structure of the Node is given below: **[2]**

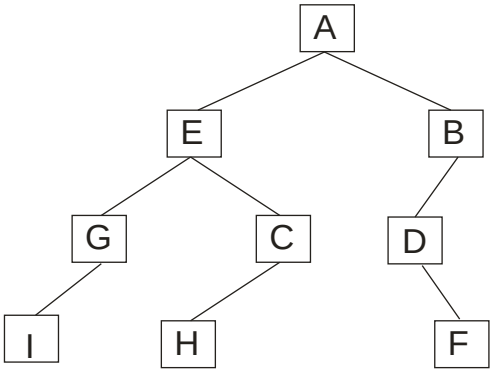
```
class   Node
{
        int num;
        Node next;
}
```

Write an *Algorithm* **OR** a *Method* to find and display the sum of even integers from an existing linked list.

The method declaration is as follows:

**void  SumEvenNode( Node str  )**

(b) Answer the following questions from the diagram of a Binary Tree given below:




(i) Write the pre-order traversal of the above tree structure. **[1]**

(ii) State the size of the tree. **[1]**

(iii) Name the siblings of the nodes E and G. **[1]**

## Comments of Examiners

(a) Majority of the candidates answered this question very well. Some candidates did not move the pointer to the next node and checked for null. Some candidates wrote the algorithm in simple English language covering all the main steps. A few candidates did not create the temporary pointer. Loop was missing in a few answer scripts.

(b) (i) Majority of the candidates answered this question very well. However, some candidates wrote the other orders instead of preorder of the tree. A few candidates did not place one or two nodes correctly.

(ii) Many candidates answered this question well. However, some candidates gave the height instead of the size.

(iii) Well answered by several candidates. However, some candidates wrote the child nodes instead of for the siblings.

### *Suggestions for teachers*

- Give adequate practice in methods / algorithms with link list and binary tree data structure.
- Train students to illustrate the link list and the binary trees using the diagrams.
- Familiarise the students with the concept of temporary pointer, checking for null condition and moving pointer to the next node giving relevant examples.
- Explain thoroughly to the students the concept of root, height, depth, size, degree, siblings, nodes (internal and external) using a binary tree diagram.

## MARKING SCHEME

## Question 12

(a) **ALGORITHM:**

Step 1. Start
Step 2. Set temporary pointer to the first node
Step 3. Repeat steps 4 and 5 until the pointer reaches null. Display sum, exit
Step 4. check for even number, if found accumulate
Step 5. Move pointer to the next node
Step 6. End algorithm

**METHOD:**

```
void SumEvenNode(Node str)
{   Node temp=new Node(str);
    int s=0;
    while(temp!=null )
     {  if (temp.num%2==0)
            s=s+temp.num;
            temp=temp.next;
       }
    System.out.println("sum of even integers="+s);
}
```

(b) (i) A E G I C H B D F

(ii) 9

(iii) SIBLING OF E IS B.
SIBLING OF G IS C.

# GENERAL COMMENTS

**Topics found difficult by candidates**

- Propositional statements for Converse and Contrapositive.
- Interface definition.
- Complexity and comparing $O(n^2)$ and $O(2^n)$.
- Output using recursive function.
- K-MAPS (grouping, redundant groups, map-rolling, place value).
- Recursive technique.
- Passing objects to functions.
- Queue operations for push() and pop() operations.
- Size of a binary tree.

**Concepts in which candidates got confused**

- Propositional statements.
- Use of Single instance variable for multiple operations in various functions.
- Link list and Queue.
- Size and Siblings of a tree

**Suggestions for candidates**

- Revise and integrate the concepts studied in Class XI with the Class XII syllabus.
- Avoid selective study.
- Give equal importance to all the topics mentioned in the syllabus.
- Practise each topic/sub–topic with as many examples as possible.
- Gather knowledge of the subject comprehensively with latest concepts and improved techniques in programming using reliable resources like text books, Wikipedia etc.
- Download directly proper definitions, output programs, algorithms etc. from the net.
- Prepare summary for each chapter or use high lighters to recognize the important terms and definitions.
- Understand all key terms/important words, definitions etc. comprehensively and learn.
- Practise on the system on a regular basis to understand the syntax and to correct the errors.
- Always write laws while reducing a Boolean Expression.
- Practise one form of K-Map with proper place value for both SOP and POS.
- Mention Documentation with each program.
- Declare the class with data members and member functions.
- Expand or define each function according to the instructions given by the side of each function.
- Practise constructors with every program.
- Understand the logic of a program instead of memorising it.
- Solve previous years' ISC question papers.
- Always show workings, wherever required, on the side of each question.
- Always answer to any question as per the requirement of the question.