

33/38

FIRST PRE-BOARD EXAMINATION 2024-25

Class XII (ISC)

COMPUTER SCIENCE

ANSWER KEY.

Maximum Marks: 70

Time Allotted: Three Hours

Reading Time: Additional Fifteen minutes

Instructions to Candidates

1. You are allowed an **additional 15 minutes** for only reading the question paper.
2. You must **NOT** start writing during reading time.
3. The question paper has **12** printed pages.
4. It is divided into **two parts** and has **11 questions** in all.
5. **Part I** is compulsory and has **two** questions. Answer **all** questions
6. **Part II** is divided into **three sections**: A, B and C.
7. While attempting **Multiple Choice Questions** in Part I, you are required to **write only ONE option as the answer**.
8. **Each section in Part II** has **three questions**. Any **two** questions have to be attempted from each section.
9. The intended marks for questions or parts of questions are given in the brackets [].

PART I – 20 MARKS

Answer all questions

While answering questions in this Part, indicate briefly your working and reasoning, wherever required.

Question 1

- (i) Simplify the Boolean expression $(A+B)(A+B')$. What is the result? [1]
- (a) A
(b) B
(c) $A + B'$
(d) $A \cdot B$

- (ii) Assertion (A): A recursive function with a missing base case can cause a stack overflow.
Reason (R): Infinite recursion occurs when a recursive function calls itself indefinitely without a stopping condition.
Choose the correct option. [1]

- (a) Both (A) and (R) are true and (R) is the correct explanation of (A).
(b) Both (A) and (R) are true and (R) is not the correct explanation of (A).
(c) (A) is true but (R) is false.
(d) (A) is false but (R) is true.
- 12
comp sc

(iii) A digital circuit for a security system has two sensors, A and B. The alarm should sound if only one sensor is triggered, but not both. Which of the following Boolean expressions represents this requirement? [1]

- (a) $A \cdot B$
- (b) $A'B + AB'$
- (c) $A + B$
- (d) $A' + B'$

(iv) Which Boolean law is represented by the expression $A + AB = A$? [1]

- (a) Idempotent Law
- (b) Absorption Law
- (c) Identity Law
- (d) Complement Law

(v) The complement of the reduced expression of $F(A,B) = \Sigma(0,1,2)$ is [1]

- (a) 1
- (b) $A \cdot B$
- (c) 0
- (d) $A' + B'$

(vi) Assertion (A): The statement $(p \wedge q) \vee (\neg p \wedge \neg q)$ is not a tautology.
Reason (R): A tautology is a proposition that is always true.
Choose the correct option. [1]

- (a) Both (A) and (R) are true and (R) is the correct explanation of (A).
- (b) Both (A) and (R) are true and (R) is not the correct explanation of (A).
- (c) (A) is true but (R) is false.
- (d) (A) is false but (R) is true.

(vii) According to the principle of Duality, the dual of the Boolean equation [1]

$$Y \cdot X + X' + 1 = 1 \text{ is}$$

- (a) $(Y'X) \cdot X' \cdot 0 = 0$
- (b) $(Y + X') + X' + 1 = 1$
- (c) $(Y + X) \cdot X' \cdot 0 = 0$
- (d) $(Y + X) + X' + 0 = 0$

(viii) Assertion(A): In java a subclass can access all public and protected members of its superclass.

Reason (R): The protected keyword allows access to members within the same package and subclasses in other packages.

Choose the correct option.

[1]

- (a) Both (A) and (R) are true and (R) is the correct explanation of (A).
- (b) Both (A) and (R) are true and (R) is not the correct explanation of (A).
- ✓(c) (A) is true but (R) is false.
- (d) (A) is false but (R) is true.

(ix) Determine the worst case complexity of the given code:

[1]

```
for ( int i=1; i<=m; i++)  
{  
    for( int j=1; j<=n; j++)  
        System.out.println(i+j);  
}
```

$O(mn)$

(x) In order to make 4x1 multiplexer, write the quantity of AND, OR and NOT gates required.

[1]

[1 mark for any two correct quantity]

Question 2

(i) Convert the following infix notation to postfix form:

[2]

$A \times (B + C) - D / E$ $ABC + * DE / -$

(ii) 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]

(iii) Consider the below code snippet and answer the following questions:

```
public class PowerExample {  
    public static int power(int base, int exp) {  
        if (exp == 0) {  
            return 1;  
        } else {  
            return base * power(base, exp - 1);  
        }  
    }  
}
```

(a) What condition causes the recursion to stop in the power method?

[1]

$if (exp == 0)$

- (b) What would power(2, 3) return when this method is called? 8 [1]
 (c) How many times is the recursive call power(base, exp - 1) made for power(3, 2)? 2 [1]

(iv) The function void prodofeven() is a member of some class. It has been used to find the product of even numbers in the array. There are three places in the code marked by ?1?, ?2?, and ?3? which must be replaced by expressions or statements so that the function works correctly.

```
int prodofeven(int a[ ], int i ) {
    if (i == a.length)
        return ?1?;
    else {
        if (?2?)
            return a[i] * prodofeven(a, i+1);
        else
            return ?3?;
    }
}
```

- (a) What is the expression or statement at ?1? [1]
 (b) What is the expression or statement at ?2? [1]
 (c) What is the expression or statement at ?3? [1]

a[i] % 2 == 0
 prodofeven(a, i+1)

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 3

- (i) A smart home system is designed to trigger an alarm based on the following conditions: [5]
- If the **Motion Sensor** and **Door Sensor** are activated,
OR
 - If the **Motion Sensor** is deactivated, but the **Door Sensor** and **Window Sensor** are activated,
OR
 - If the **Door Sensor** is deactivated, but the **Motion Sensor** and **Smoke Detector** are activated,
OR
 - When all sensors are activated.

The Inputs are:

M	Motion Sensor is activated
D	Door Sensor is activated
W	Window Sensor is activated
S	Smoke Detector is activated

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

Output: X: Alarm status (1 indicates alarm triggered, 0 indicates no alarm)

Draw the truth table for the inputs and outputs given above and write the SOP expression for X(M,D,W,S)

*4 for truth table + 1 for SOP
* Expression in the answer key is wrong,
cardinal is correct.*

(ii) (a) Reduce the above expression X(M,D,W,S) by using a 4-variable Karnaugh map, showing the various groups (i.e., octals, quads, and pairs). [4]

(b) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]

[1 for each group + 1 for K-Map]

Question 4

(i) Given the Boolean function:

$$F(A, B, C, D) = (A+B+C+D). (A+B+C+\bar{D}). (A+B+\bar{C}+D). (A+\bar{B}+C+\bar{D}). (A+\bar{B}+\bar{C}+\bar{D})$$

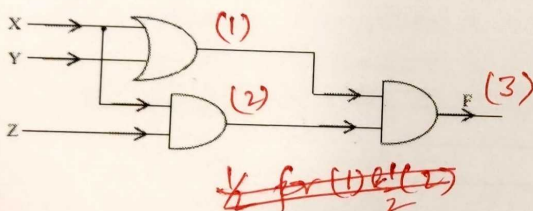
$$(\bar{A}+B+C+D). (\bar{A}+B+\bar{C}+D). (\bar{A}+\bar{B}+\bar{C}+D)$$

- (a) Reduce the above expression by using a 4-variable Karnaugh map, showing the various groups (i.e., octals, quads, and pairs). *1 for each group* [4]
 (b) Draw the logic gate diagram for the reduced expression using NOR gates only. Assume that the variables and their complements are available as inputs. [1]

(ii) Illustrate the logic circuit for a binary to octal decoder. Explain its operation. [4+1]

Question 5

(i) 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 [5]



1/2 for (1) + 1/2 for (2)
1 for (3) + 1 for simplification
+ 1 for Name + 1 for logic gate

(ii) Using a truth table, state whether the following proposition is a Tautology, Contradiction or Contingency:

$$\sim(P \rightarrow Q) \Leftrightarrow (R \vee Q)$$

1/2 (P → Q) 1/2 for final [3]
1/2 (P → Q) 1 for final statement
1/2 (R ∨ Q)

(iii) (a) Draw the circuit of a two-input XOR gate using NAND gates. [1]

(b) Write truth table and expression of sum and carry for half adder. [1]

[1 for either truth table or expression]

SECTION B

Each program should be written in such a way that it clearly depicts the logic of the problem. This can be achieved by using mnemonic names and comments in the program.

(Flowcharts and Algorithms are **not** required.)

The programs must be written in Java.

Question 6

Design a class StringOperations to perform various operations on a given string. [The class should be able to reverse the string, count the number of vowels, and convert the string to uppercase.]

[10]

Example: For the input string "Hello World":

- Reversed: "dlroW olleH"
- Vowel Count: 3
- Uppercase: "HELLO WORLD"

Some of the members of the class are given below:

Class name		: StringOperations
Data members/Instance Variables:	}	
inputString		① : to store the input string
vowelCount		: to store the count of vowels in the string
Methods/Member functions:		
StringOperations(String str)	①	: parameterized constructor to initialize the data member inputString=str
String reverseString()	②	: returns the reversed version of inputString
int countVowels()	②	: updates the vowelCount and returns the count of vowels in inputString
String toUpperCase()	①	: returns the uppercase version of inputString
void displayResults()	①	: displays the results of the operations: reversed string, vowel count, and uppercase string

Specify the class StringOperations giving details of the constructor, String reverseString(), int countVowels(), String toUpperCase(), and void displayResults(). Define a main() function to create an object and call the functions accordingly to enable the task.

1 for main() + 1 for VDC

Question 7

Design a class MAT to find the sum of digits of each element of the matrix and store the sum into another matrix. [10]

Example:

23	439	68
7	59	129
100	6	35

becomes

5	16	14
7	14	12
1	6	8

Some of the members of the class are given below:

Class name

: MAT

Data members/Instance Variables:

Arr[][]

① : to store integer elements

m

: to store the number of rows

n

: to store the number of columns

Methods/Member functions:

MAT(int mm, int nn)

①

: parameterized constructor to initialize the data
Members m=mm and n=nn.

void fillarray()

①

: to enter elements in the array

int sumDigit(int x)

②

: return the sum of digit of x

void store (MAT P)

②

: to store sum of digit of each number present in array
of object P in the array of current object.

void display()

①

: displays the array element in matrix form.

Define the class MAT giving details of the constructor, void fillarray(), int sumDigit(int), void store(MAT) and void display(). Define the main() function to create objects and call the functions accordingly to enable the task.

1 for main()
1 for VDC

Question 8

Design a class FibonacciCalculator to compute the Fibonacci sequence up to a given number using recursion. [The class should be able to calculate the Fibonacci number at a specific position and display the entire sequence up to that position.] [10]

Example: For the input position 7:

- Fibonacci at position 7: 8
- Sequence up to position 7: 0, 1, 1, 2, 3, 5, 8

Some of the members of the class are given below:

- Class name** : FibonacciCalculator
- Data members/Instance Variables:**
- position : to store the input position in the Fibonacci sequence
 - fibonacciResult : to store the Fibonacci number at the specified position
- Methods/Member functions:**
- FibonacciCalculator(int pos) ① : parameterized constructor to initialize the data member position = pos
 - int calculateFibonacci(int n) ③ : returns the Fibonacci number at position n using recursion
 - void displayFibonacciSequence() ② : displays the entire Fibonacci sequence up to the specified position
 - void displayResults() ① : displays the results of the operations: Fibonacci number at the given position and the sequence

Specify the class FibonacciCalculator giving details of the constructor, int calculateFibonacci(int n), void displayFibonacciSequence(), and void displayResults(). Define a main() function to create an object and call the functions accordingly to enable the task.

Handwritten note:
1 for main() + 1 for VDC

SECTION C

Answer **any two** questions.

Each program should be written in such a way that it clearly depicts the logic of the problem stepwise. This can be achieved by using comments in the program and mnemonic names or pseudo codes for algorithms. The programs must be written in Java and the algorithms must be written in general / standard form, wherever required / specified.
(Flowcharts are **not** required.)

Question 9

- (i) Circular queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. Define a class cqueue with the following details. [4]

Some of the members of the class are given below:

Class name : cqueue

Data members/Instance Variables:

ele[] : array to hold the integer elements.
cap : store the maximum capacity of the element.
front : to point the index of the front.
Rear : to point the index of the rear.

Methods/Member functions:

cqueue(int max) : constructor to initialise the data members cap=max, front=0, rear=0 and create the integer array.
void insert(int v) **2** : to add integer from the rear index in a circular manner if possible else display the message("queue is full")
int delete() **2** : to remove and return element from front end if any else return -999
void display() : to displays the elements of circular queue

Specify the class cqueue giving details of void insert(int) and int delete (). Assume that the other functions have been defined. The main () function and algorithm need not to be written.

- (ii) What do you understand by overflow and underflow in stacks?

[1]

Question 10

An interface Employee is defined with a method calculateSalary() which calculate the salary of the employee. A super class Person has been defined to store the name and age of a person. Define a sub class Manager which uses the properties of the interface Employee and the class Person to calculate and display the gross salary of the employee. The details of the members of the interface and both the classes are given below: [5]

Interface name : Employee

Member functions/methods:

double calculateSalary()

class name : Person

Data member/instance variables:

name : to store the name

age : to store the age

Member function/methods:

Person(..) : parameterized constructor to initialize the data

Members

void display () : display the details

Class name

Data member/instance variable

Basic : to store the basic pay in decimal.

Member functions/methods:

Manager(...) : parametrized constructor to initialize the members of both the classes

① super() should be the first statement.

double calculateSalary () : to calculate and return the gross salary of the employee as per the following condition:

basic	HRA	DA
<=5000	12% of basic	30% of basic
>5000	15% of basic	35% of basic

②

Gross salary = basic + HRA + DA

void display () : display all the details along with gross salary.

Assume that the interface Employee and the super class Person has been defined. Using the concept of inheritance, specify the class Manager giving details of the constructor (...), double calculateSalary() and void display(). The interface, superclass, main function and algorithm need not to be written.

① super keyword 1/2 marks

1/2 for extends, 1/2 for implement

Question 11

(i) Consider a linked list created from the objects of the class Node, defined as:

[2]

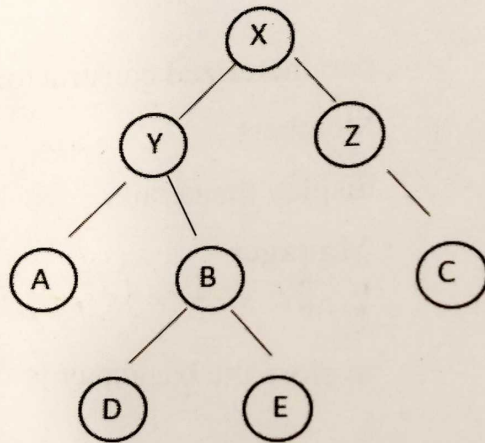
```
class Node {  
    int n;  
    Node next;  
}
```

Write an algorithm or a method to **count nodes with values greater than a given integer x**.
The method declaration is as follows:

```
int CountGreaterThan(Node head, int x)
```

1 for loop + 1 for logic

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



- (a) Write the **pre-order traversal** of the left subtree of the above structure. *YABDE* [1]
- (b) State the **degree** of the Nodes X and B. *2, 2* [1]
- (c) Mention the **internal nodes** of the right subtree. *Z* [1]