Spondon Ganguli
An Educator, Artist & Author

ISC 2024 EXAMINATION SPECIMEN QUESTION PAPER COMPUTER SCIENCE

SOLUTION

Prepared by Spondon Ganguli

*This Paper is divided into **two** Parts – Part I and Part II.*

All** questions from **Part I** are to be attempted**.

Part II** is divided into **three** Sections – Section A, Section B and Section C**.

Any two** questions to be attempted from all three **Sections.

PART - I

(Attempt ***all*** *questions from this part.)*

Question 1.

(i)	Absorption law states that:	
	(a)	A • (A' + B) = A
	(b)	A + (A • B) = A
	(c)	Both (a) and (b)
	(d)	A • (B + C) = A • B + A • C

Answer. (b) A + (A.B) = A
(Explanation: By Absorption law, adding or multiplying a term with a disjunction or conjunction of itself does not change the value of the item)

(ii)	Assertion : A=0 B=1 C=0 and D=1 and minterm is A'•B•C'•D Reason : The final sum term must be 0 so A and C are complemented.	
	Which one of the following options is correct?	
	(a)	Both Assertion and Reason are true, and Reason is the correct explanation for Assertion.
	(b)	Both Assertion and Reason are true, but Reason is not the correct explanation for Assertion.
	(c)	Assertion is true and Reason is false.
	(d)	Assertion is false and Reason is true.

Answer. (c) Assertion is true and Reason is false
Explanation: In Boolean Algebra, for minterm, every term value must be 1, not 0.

(iii)	According to the Principle of duality, the Boolean equation (P + Q') • R • 1 = P • R + Q' • R will be equivalent to:	
	(a)	P • Q' + R + 1 = (P + R) • (Q' + R)
	(b)	P • Q' + R + 0 = (P + R) • (Q' + R)
	(c)	P' • Q + R + 1 = (P' • R') • (Q + R')
	(d)	P • Q' + R • 0 = (P + R) • (Q' + R)

Answer. (b) P • Q' + R + 0 = (P + R) • (Q' + R)
Explanation: The principle of duality states that all ANDs will be changed to ORs and vice versa, all 0s will be changed to 1s, but the complement remains unchanged.

(iv)	The complement of the Boolean expression (X • Y)′ + Z′ is:	
	(a)	(X + Y) • Z
	(b)	X • Y • Z
	(c)	(X' + Y') • Z'
	(d)	(X' + Y') • Z

Answer. (b) X.Y.Z
Working: ((X.Y)′ + Z′)′ = (X.Y)′′ . Z′′ = X.Y.Z

(v)	The equivalent of P ^ Q ˅ ~ P ^ ~ Q will be:	
	(a)	((P ^ Q) ˅ ~ P) ^ ~ Q
	(b)	(P ^ Q) ˅ (~ P ^ ~ Q)
	(c)	P ^ (Q ˅ ~ P) ^ ~ Q
	(d)	P ^ (Q ˅ (~ P ^ ~ Q))

Answer. (b) (P ^ Q) ˅ (~ P ^ ~ Q)
Explanation: ^ means AND and ˅ means OR, and in Boolean Algebra, AND operation has higher precedence over OR operation

(vi)	Assertion : Boolean algebra and Binary number systems are different from each other. Reason : There are some basic operations like AND, OR and NOT which are performed only in Boolean algebra.	
	(a)	Both Assertion and Reason are true, and Reason is the correct explanation for Assertion.
	(b)	Both Assertion and Reason are true, but Reason is not the correct explanation for Assertion.
	(c)	Assertion is true and Reason is false.
	(d)	Assertion is false and Reason is true.

Answer. (c) Assertion is true and Reason is false.
Explanation: AND, OR and NOT operations can also be performed on Binary Numbers. Also, there are bitwise (AND, OR, NOT and XOR) operations in programming.

(vii) What is the relevance of the keyword *static* for a data member of a class?

Answer. Data members declared with 'static' keyword become a class member and there exists only one copy of that in the memory, which is being shared by all the objects of that class.

(viii) State any one purpose of using *interfaces* in Java programming.

Answer. Interfaces are used to implement multiple inheritance in Java programming.

(ix) Define *Canonical form* of an expression with respect to its *Cardinal form*

Answer. A canonical form of a Boolean expression is the logical sum of some min terms or the logical product of some max terms.

(x) State *any one* application each of *half adder* and *full adder*.

Answer. Half adder adds two binary bits, whereas full adder adds three binary bits.

Question 2	
(i)	Convert the following infix notation to postfix form. (A / B + C) / (D * (E – F)

Answer.

((A / B) + C) / (D * (E – F))
(AB/ + C) / (D * EF-)
(AB/C+) / (DEF-*)
AB/C+DEF-*/

Spondon Ganguli

An Educator, Artist & Author

(ii)	An array ARR[−4 …..6, −2…..12] , stores elements in Row Major Wise, with the address ARR[2][3] as 4142. If each element requires 2 bytes of storage, find the Base address.

Answer:

Formula for Row Major Order: $A = B + W \times ((I_A - I_B) \times C + (J_A - J_B))$

Given $A = 4142$, $W = 2$, $C = 12 - (-2) + 1 = 15$, $I_A = 2$, $J_A = 3$, $I_B = -4$, $J_B = -2$

$B = 4142 - 2 \times ((2+4) \times 15 + (3+2))$
$= 4142 - 190$
$= 3952$

(iii)

The following functions are a part of some class:

```
void Try(char ch[], int x)
{
   System.out.println(ch);
   char temp;
   if ( x<ch.length/2)
  {
     temp=ch[x];
     ch[x]= ch[ch.length-x-1];
     ch[ch.length-x-1] = temp;
     Try(ch,x+1);
   }
}
void Try1(String n)
{
  char c[]=new char[n.length()];
   for(int i=0;i<c.length;i++)
      c[i] = n.charAt(i);
  Try(c,0);
}
```

(iii)	(a)	What will the output of Try( ) when the value of ch[]={‘P’, ‘L’,‘A’, ‘Y’} and x=1?

Answer.

Try(ch[]={‘P’, ‘L’,‘A’, ‘Y’}, x=1)

PLAY

x=1, ch.length/2 = 2, 1<2 → temp=L ch[1]=A → ch[2]=L

Try(ch[]={‘P’, ‘A’,‘L’, ‘Y’}, x=2)

PALY

x=2, ch.length/2=2, 2=2 → recursion ends

OUTPUT:

```
PLAY
PALY
```

(iii)	(b)	What will the output of Try1( ) when the value of n=”SKY”?

Answer.

```
Try1( )
        i=0 → i<3 → c[0]=’S’
        i=1 → i<3 → c[1]=’K’
        i=2 → i<3 → c[2]=’Y’
        c[ ]={‘S’,’K’,’Y’}
        Try(c[ ]={‘S’,’K’,’Y’},x=0)
                SKY
                x=0, ch.length/2 = 1, 0<1 → temp=S ch[0]=Y → ch[1]=S
        Try(c[ ]={‘Y’,’K’,’S’},x=1)
                YKS
                x=1, ch.length/2=1, 1=1 → recursion ends
OUTPUT
  SKY
  YKS
```

(iv) The following function is a part of some class which computes and returns the value of a number ‘p’ raised to the power ‘q’ (p^q). There are some places in the code marked by ?1? , ?2? , ?3? which must be replaced by an expression / a statement sothat the function works correctly.

```
double power ( double p , int q )
{  double r = ?1? ;
   int c = ( q<0 ) ? -q : q ;
   if ( q == 0)
                return  1 ;
   else
   {
        for (int i = 1; i <= c ;?2?, i++);
         return (q>0)? r : ?3?;
   }
}
```

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

Answer.

(a) 1
(b) r = r*p
(c) 1.0/r

PART - II

SECTION A

*(Attempt **any two** questions from this **Section**.)*

Spondon Ganguli
An Educator, Artist & Author

Question 3

(i) A Football Association coach analyses the criteria for a win/draw of his team depending on the following conditions.

(i)If the Centre and Forward players perform well but Defenders do not perform well

OR

(i)If Goalkeeper and Defenders perform well but the Centre players do not perform well.

OR

(i)If all perform well.

The inputs are:

INPUTS	
C	Centre players perform well
D	Defenders perform well
F	Forward players perform well
G	Goalkeeper perform well

(In all the above cases, 1 indicates yes and 0 indicates no.)
Output: **X** - Denotes the win/draw criteria [1 indicates win/draw and 0 indicates defeat in all cases]
Draw the truth table for the inputs and outputs given above and write the
SOP expression for **X(C, D, F, G)**.

Answer.

Condition 1: C (Centre players) . D’ (Defenders) . F (Forward players) , where C = 1, D = 0 and F = 1

Condition 2: C’(Center players) . D (Defenders) . G (Goalkeeper), where C = 0, D = 1, F = 1

Condition 3: C.D.F.G , where C = 1, D = 1, F = 1, G = 1

Truth table for **X(C, D, F, G)**:

C	D	F	G	X	Minterm	Design
0	0	0	0			0
0	0	0	1			1
0	0	1	0			2
0	0	1	1			3
0	1	0	0			4
0	1	0	1	1	C’.D.F’.G	5
0	1	1	0			6
0	1	1	1	1	C’.D.F.G	7
1	0	0	0			8
1	0	0	1			9
1	0	1	0	1	C.D’.F.G’	10
1	0	1	1	1	C.D’.F.G	11
1	1	0	0			12
1	1	0	1			13
1	1	1	0			14
1	1	1	1	1	C.D.F.G	15

X(C, D, F, G) = C’.D.F’.G + C’.D.F.G + C.D’.F.G’ + C.D’.F.G + C.D.F.G

$= \sum (5, 7, 10, 11, 15)$

(ii) Reduce the above expression **X (C, D, F, G)** by using a 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs.

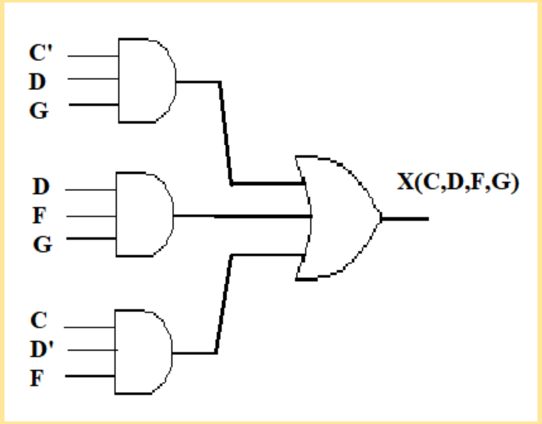
Answer. K-MAP for X(C,D,F,G) = ∑ (5, 7, 10, 11, 15)

	F'.G'	F'.G	F.G	F.G'
C'.D'	0	1	3	2
C'.D	4	1 5	1 7	6
C.D	12	13	1 15	14
C.D'	8	9	1 11	1 10

Pair 1 (m5, m7) = C'.D.G
Pair 2 (m7, m15) = D.F.G
Pair 3 (m11, m10) = C.D'.F

X(C, D, F, G) = C'.D.G + D.F.G + C.D'.F

Logic diagram:



Question 4		
(i)	(a)	Reduce the Boolean function F(A,B,C,D) = π (0, 1, 2, 3, 4, 6, 9, 11, 13) by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs).

Answer. K-MAP

	C+D	C+D’	C’+D’	C’+D
A+B	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
A+B’	0 $_{4}$	$_{5}$	$_{7}$	0 $_{6}$
A’+B’	$_{12}$	0 $_{13}$	$_{15}$	$_{14}$
A’+B	$_{8}$	0 $_{9}$	0 $_{11}$	$_{10}$

Quad 1 (m0,m4,m2,m6) = A+D

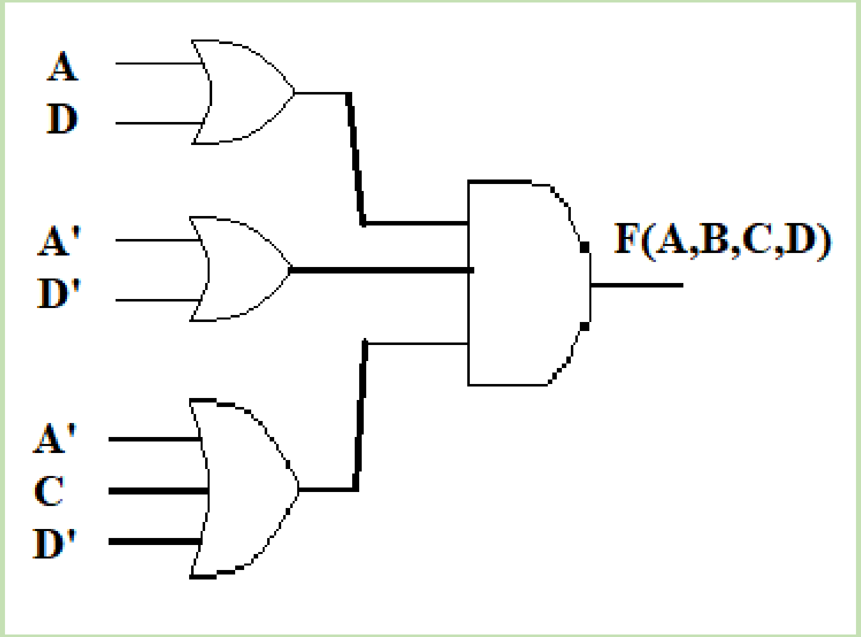
Quad 2 (m1,m3,m9,m11) = B+D’

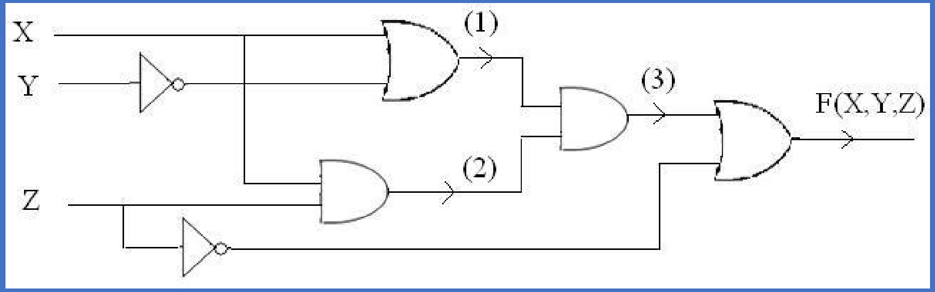
Pair 1 (m13,m9) = A’+C+D’

F(A,B,C,D) = (A+D).(B+D’).(A’+C+D’)

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

Answer. F(A,B,C,D) = (A+D).(A'+D').(A'+C+D')



(ii)	(a)	From the logic circuit diagram given below, name the outputs (1), (2) and (3) and finally derive the Boolean expression (F) and simplify it. Identify the propositional connective which is equivalent to the simplified Boolean expression.
		
	(b)	If A=1 and B=0 then find the value of (A' + 1) • B

Spondon Ganguli
An Educator, Artist & Author

Answer.

(a) **1. X+Y'**

2. X.Z

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

F(X,Y,Z) = (X+Y').X.Z + Z'

(X+Y').X.Z + Z'
= X.X.Z + X.Y'.Z + Z'
= X.Z + X.Y'.Z + Z'
= X.Z (1 + Y') + Z'
= X.Z + Z'
= X+Z'

(b) **(A' + 1) . B = (0 + 1). 0 = 0**

Question 5

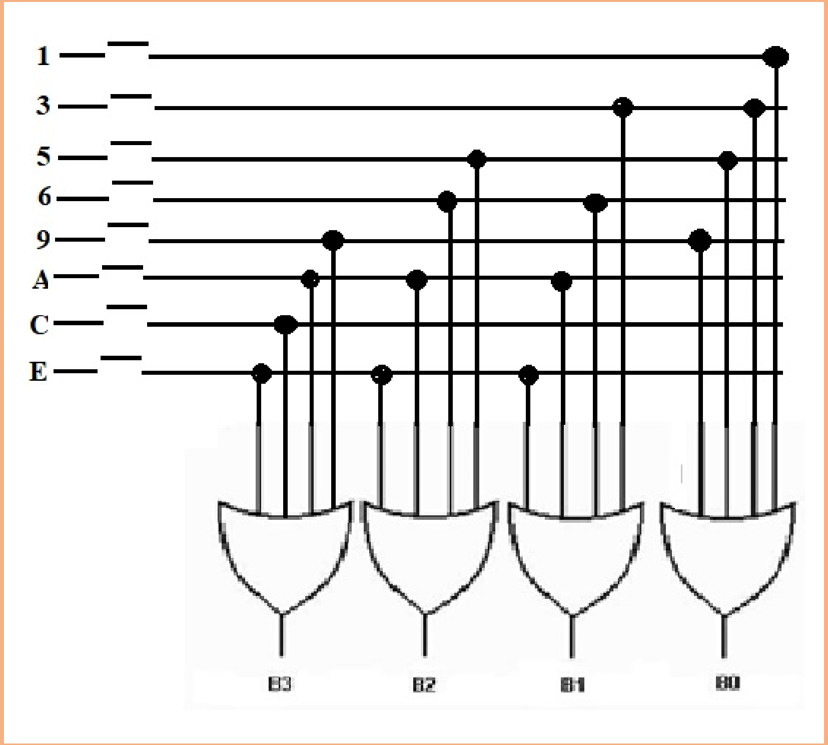
(i)	Draw the logic circuit to encode the following Hexadecimal number (1, 3, 5, 6, 9, A, C, E) to its binary equivalents. Also state the binary equivalents ofthe given numbers.

Answer. Hexadecimal to Binary:

Truth Table

HEXADECIMAL (INPUT)	BINARY (OUTPUT)			
	B_3	B_2	B_1	B_0
1	0	0	0	1
3	0	0	1	1
5	0	1	0	1
6	0	1	1	0
9	1	0	0	1
A	1	0	1	0
C	1	1	0	0
E	1	1	1	0

Logic Diagram



(ii)	Verify if the following proposition is valid using the truth table: A => (B A C) = (A => B) A (B => C)
(iii)	How is a 2 to 4 decoder related to 4:1 multiplexer?

Answer.

(ii) A => (B A C) = (A => B) A (B => C)

INPUT				L.H.S			R.H.S
A	B	C	B.C	**A→B.C**	A→B	B→C	**(A→B).(B→C)**
0	0	0	0	**1**	1	1	**1**
0	0	1	0	**1**	1	1	**1**
0	1	0	0	**1**	1	0	**0**
0	1	1	1	**1**	1	1	**1**
1	0	0	0	**0**	0	1	**0**
1	0	1	0	**0**	0	1	**0**
1	1	0	0	**0**	1	0	**0**
1	1	1	1	**1**	1	1	**1**

Since L.H.S. != R.H.S, thus we can say the proposition is invalid.

(iii) Multiplexer and decoder are combinational digital circuits extensively used to transfer signals in many communication systems. Basically, the multiplexer and decoder both perform almost identical functions. Now, 2 to 4 decoder works on 2 input lines that get transferred to 4 output lines and in a similar way in 4:1 multiplexer, 2 select lines allow one of the several input signals to be passed through 4 AND gates that further pass through a single OR gate. Thus, 2 to 4 decoders and 4:1 multiplexers have the same number of AND gates through which the input lines are passed for final selection.

SECTION B

*(Attempt **any two** questions from this **Section**.)*

Question 6

An Evil Number is a number which contains even number of 1's in its binaryequivalent.
Example: Binary equivalent of 10 = 1010 which contains even number on 1's.
Thus, 10 is an Evil Number.
Design a class Evil to check if a given number is an Evil number or not. Some of themembers of the class are given below: .

Class name	:	Evil
Data members/instance variables:		
num	:	to store a positive integer number
bin	:	to store the binary equivalent
Methods / Member functions:		
Evil( )	:	default constructor to initialize the data memberwith legal initial value
void acceptNum( )	:	to accept a positive integer number
void rec_bin (int x)	:	to convert the decimal number into its binaryequivalent using recursive technique
void check( )	:	to check whether the given number is an Evil number by invoking the function rec_bin() and to display the result with an appropriatemessage

Specify the class Evil giving details of the constructor( ), void acceptNum( ), void rec_bin(int) and void check( ). Define a main( ) function to create an object and call all the functions accordingly to enable the task.

Solution:

```
import java.util.Scanner;
class Evil
{
   int num, bin;
   Evil()//default constructor
   {
      num=bin=0;
   }
   void acceptNum()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a positive number");
      num=Math.abs(sc.nextInt());
   }
   void rec_bin(int x)
   {
      if(x==0)
       return;//ends the recursion and sends back the control
      else
      {
         int r=x%2;//getting the remainder
         rec_bin(x/2);//recursive function call
         bin=bin*10+r;//forming the binary number
      }

   }
```

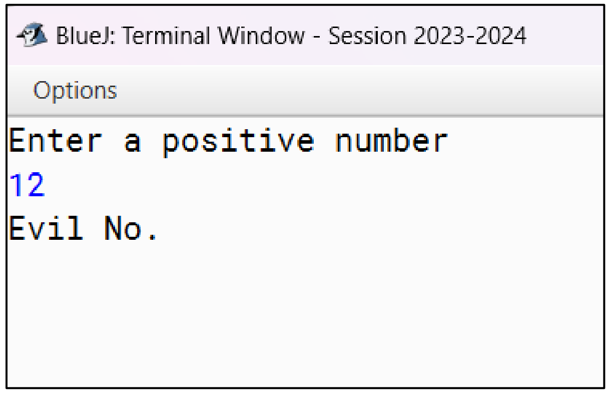
How is the recursive function working?

Rec_bin(10) //x=10
R=10%2=0
Rec_bin(10/2=5) //x=5
R=5%2=1
Rec_bin(5/2=2) //x=2
R=2%2=0
Rec_bin(2/2=1) //x=1
R=1%2=1
Rec_bin(1/2=0) //x=0
Recursion ends here
And backtracking begins
Returned by Rec_bin(0), Bin=10*0=0
Returned by Rec_bin(1), Bin=10*0+1=1
Returned by Rec_bin(2), Bin=1*10+0=10
Returned by Rec_bin(5), Bin=10*10+1=101
Returned by Rec_bin(10), Bin=101*10+0=1010

Solution (contd.)

```
void check()
    {
        int count=0;
        rec_bin(num); //calling the recursive function
        for(int m=bin; m>0; m=m/10)
            count+=m%10;//adding all 1s from bin to count
        if(count%2==0)//total no. of 1 is even
            System.out.println("Evil No.");
        else
            System.out.println("Not a Evil No.");
    }
    public static void main(String args[])
    {
        Evil eobj=new Evil(); //object creation
        eobj.acceptNum(); //calling the method through object
        eobj.check();
    }
}
```

```
int m=bin;
while(m>0)
{
   int r=m%10;
   if(r==1)
     count ++;
   m=m/10;
}
```



```
Enter a positive number
12
Evil No.
```

Question 7

A class Composite contains a two-dimensional array of order [m x n]. The maximum values possible for both ‘m’ and ‘n’ is 20. Design a class Composite to fill the array with the first (m x n) composite numbers in column wise.
[Composite numbers are those which have more than two factors.]
The details of the members of the class are given below:

Class name	:	Composite
Data members/instance variables:		
arr[ ] [ ]	:	integer array to store the composite numbers column wise
m	:	integer to store the number of rows
n	:	integer to store the number of columns
Member functions/methods:		
Composite(int mm, int nn)	:	to initialize the size of the matrix, m=mm and n=nn
int isComposite(int p)	:	to return 1 if the number is composite otherwise returns 0
void fill ( )	:	to fill the elements of the array with the first (m × n) composite numbers in column wise
void display( )	:	to display the array in a matrix form

Specify the class Composite giving details of the constructor(int,int), int isComposite(int), void fill() and void display( ). Define a main( ) function to create an object and call all the functions accordingly to enable the task.

Solution:

```
import java.util.Scanner;
class Composite
{
   int arr[][];
   int m, n;
   Composite(int mm, int nn)//parameterized constructor
   {
      m=mm;
      n=nn;
      arr=new int[m][n];//memory allocation for the array
   }
   int isComposite(int p)
   {
      if(p==1||p==2||p==3||p==5||p==7)//the no. is one or a prime digit
         return 0;
      else if(p%2==0||p%3==0||p%5==0||p%7==0)//all non prime nos.
         return 1;
      else//all prime nos.
         return 0;

   }
```

```
//alternative composite no. checking
int c=0;
for(int i=1;i<=p;i++)
{
  if(p%i==0)
    c++;
}
if(c==1||c==2)
  return 0;
else
  return 1;
```

Solution (contd.)

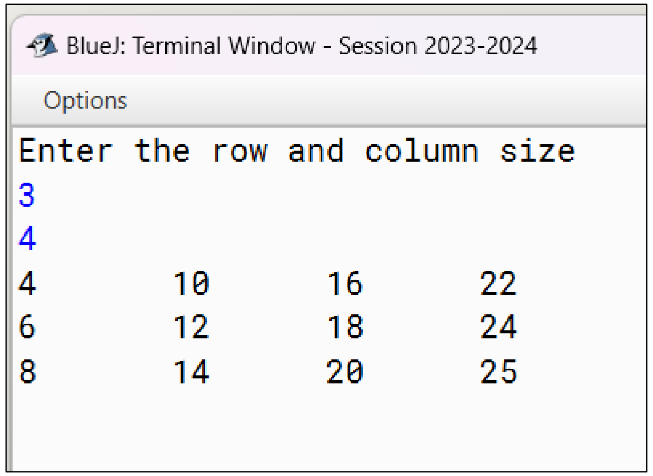
```
void fill()
{
   int i=0, j=0, a=1;
   do
   {
      if(i<m)
      {
         if(isComposite(a)==1) //composite no
            arr[i++][j]=a; //at arr[i][j] storing the next composite no.
      }
      else
      {
         j++; //moving to the next column
         i=0; //first row of the new column
      }
      a++; //next natural number generated
   }while(j<n); //loop continues until all the columns completed
}
```

//alternate logic to store the next
//composite number in the array

```
int i, j, a=1;
for (j=0; j<n; j++)
{
  for(i=0; i<m;  )
  {
     if(isComposite(a)==1)
        arr[i++][j]=a;
     a++;
   }
}
```

Solution (contd.)

```
void display()
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
            System.out.print(arr[i][j]+"\t");
        System.out.println();
    }
}
public static void main(String ar[])
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the row and column size");
    int a=sc.nextInt();
    int b=sc.nextInt();
    Composite obj=new Composite(a,b);
    obj.fill();
    obj.display();
}
}
```



```
Enter the row and column size
3
4
4       10      16      22
6       12      18      24
8       14      20      25
```

Question 8

A class Encode has been defined to replace only the vowels in a word by the nextcorresponding vowel and form a new word.

i.e. A E, E à I, I à O, O à U, U à A and

a à e, e à i, i à o, o à u, and u à a

Example: Input: Institution

Output: Onstotatoun

Some of the members of the class are given below:

Class name	:	Encode
Data members/instance variables:		
word	:	to store a word
length	:	integer to store the length of the word
new_word	:	to store the encoded word
Methods / Member functions:		
Encode( )	:	default constructor to initialize data memberswith legal initial values
void acceptWord( )	:	to accept a word
void nextVowel( )	:	to replace only the vowels from the word stored in 'word' by the next corresponding vowel and to assign it to 'newword', with the remaining alphabets unchanged
void display( )	:	to display the original word along with theencrypted word

Specify the class Encode giving details of the constructor( ), void acceptWord( ), void nextVowel( ) and void display( ). Define a main ( ) function to create an object andcall the functions accordingly to enable the task.

Solution:

```
import java.util.Scanner;
class Encode
{
    String word, new_word;
    int length;
    Encode()
    {
        word=new_word="";
    }
    void acceptWord()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a word");
        word=sc.next();
        length=word.length();
    }
```

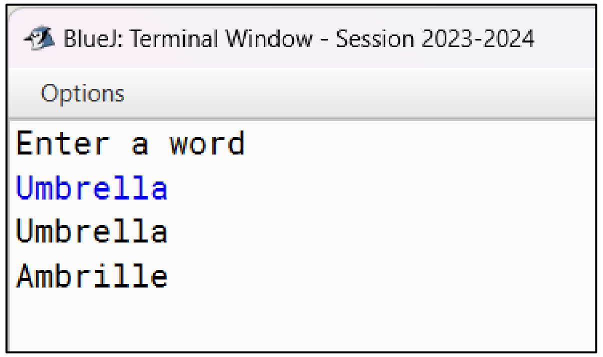
Solution (contd.)

```
void nextVowel()
{
    for(int i=0;i<length;i++)
    {
        char ch=word.charAt(i);//each letter extracted out from the word
        switch(ch)//vowels extracted from the word
        {
            case 'A':ch='E'; break;
            case 'E':ch='I'; break;
            case 'I':ch='O'; break;
            case 'O':ch='U'; break;
            case 'U':ch='A'; break;
            case 'a':ch='e'; break;
            case 'e':ch='i'; break;
            case 'i':ch='o'; break;
            case 'o':ch='u'; break;
            case 'u':ch='a'; break;
            default:
        }
        new_word+=ch;
    }
}
```

```
if(ch=='A')
ch='E';
else if(ch=='E')
ch='I';
else if(ch=='I')
ch='O';
else if(ch=='O')
ch='U';
else if(ch=='U')
ch='A';
if(ch=='a')
ch='e';
else if(ch=='e')
ch='i';
else if(ch=='i')
ch='o';
else if(ch=='o')
ch='u';
else if(ch=='u')
ch='a';
```

Solution (contd.)

```
    void display()
    {
        System.out.println(word+"\n"+new_word);
    }
    public static void main(String ar[])
    {
        Encode Eobj=new Encode();
        Eobj.acceptWord();
        Eobj.nextVowel();
        Eobj.display();
    }
}
```



```
Enter a word
Umbrella
Umbrella
Ambrille
```

SECTION C

(Attempt ***any two*** *questions from this* ***Section****.)*

Question 9

Shelf is a kind of data structure which can store elements with the restriction that an elementcan be added from the rear end and removed from the front end only.

The details of the class Shelf are given below:

Class name		:	Shelf
Data members/instance variables:			
	ele[ ]	:	array to hold decimal numbers
	lim	:	maximum limit of the shelf
	front	:	to point the index of the front end
	rear	:	to point the index of the rear end
Methods / Member functions:			
	Shelf(int n)	:	constructor to initialize lim=n, front= 0 andrear=0
	void pushVal(double v)	:	to push decimal numbers in the shelf at the rear end if possible, otherwise display the message“ SHELF IS FULL ”
	double popVal( )	:	to remove and return the decimal number fromthe front end of the shelf if any, else returns –999.99
	void display( )	:	to display the elements of the shelf
(i)	Specify the class Shelf giving details of the functions void pushVal(double) and double popVal( ). Assume that the other functions have been defined. The main( ) function and algorithm need NOT be written.		
(ii)	**Name the entity described above and state its principle.**		

Solution:

```
class Shelf
{
        double ele[];
        int lim, front, rear;
        Shelf( int n)
        {
                lim=n;
                front=rear=0;
        }
        void pushVal(double v) //insert operation in the queue
        {
                if(rear==lim-1) //if rear reached the last index
                {
                        System.out.println("SHELF IS FULL");
                        return; //returns the control back
                }
                ele[rear++]=v; //add one element and move the rear to next index position
        }
```

Solution (contd.)

```
                double popVal( ) //remove one element from the queue
                {
                        if(front==rear) //when front reached the rear it means queue is empty
                        {
                                front=rear=0; //resetting front and rear to 0
                                return -999.99;
                        }
                        return ele[front++]; //removing one element and moving front to next index
                }
                void display( )
                {
                        if(front==rear)
                                System.out.println("Empty queue");
                        else
                        {
                                for(int i=front;i<rear;i++)
                                        System.out.println(ele[i]);
                        }
                }
        }
```

(ii) The above entity is Queue and the principle implemented is First In First Out (FIFO).

Question 10

A super class Circle has been defined to calculate the area of a circle. Define a subclass Volume to calculate the volume of a cylinder.

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

Class name	:	**Circle**
Data members/instance variables:		
radius	:	to store the radius in decimals
area	:	to store the area of a circle
Methods / Member functions:		
Circle(...)	:	parameterized constructor to assign values to thedata members
void cal_area()	:	calculates the area of a circle (πr^2)
void display( )	:	to display the area of the circle
Class name		**Volume**
Data members/instance variables:		
height	:	to store the height of the cylinder in decimals
volume	:	to store the volume of the cylinder in decimals
Methods / Member functions:		
Volume(...)	:	parameterized constructor to assign values to thedata members of both the classes
double calculate( )	:	to calculate and return the volume of the cylinder using the formula ($\pi r^2 h$) where, r is the radius andh is the height
void display( )	:	to display the area of a circle and volume of acylinder

Assume that the super class Circle has been defined. Using the concept of inheritance, specify the class Volume giving the details of the constructor(...), double calculate( ) and void display( ).

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

Solution:

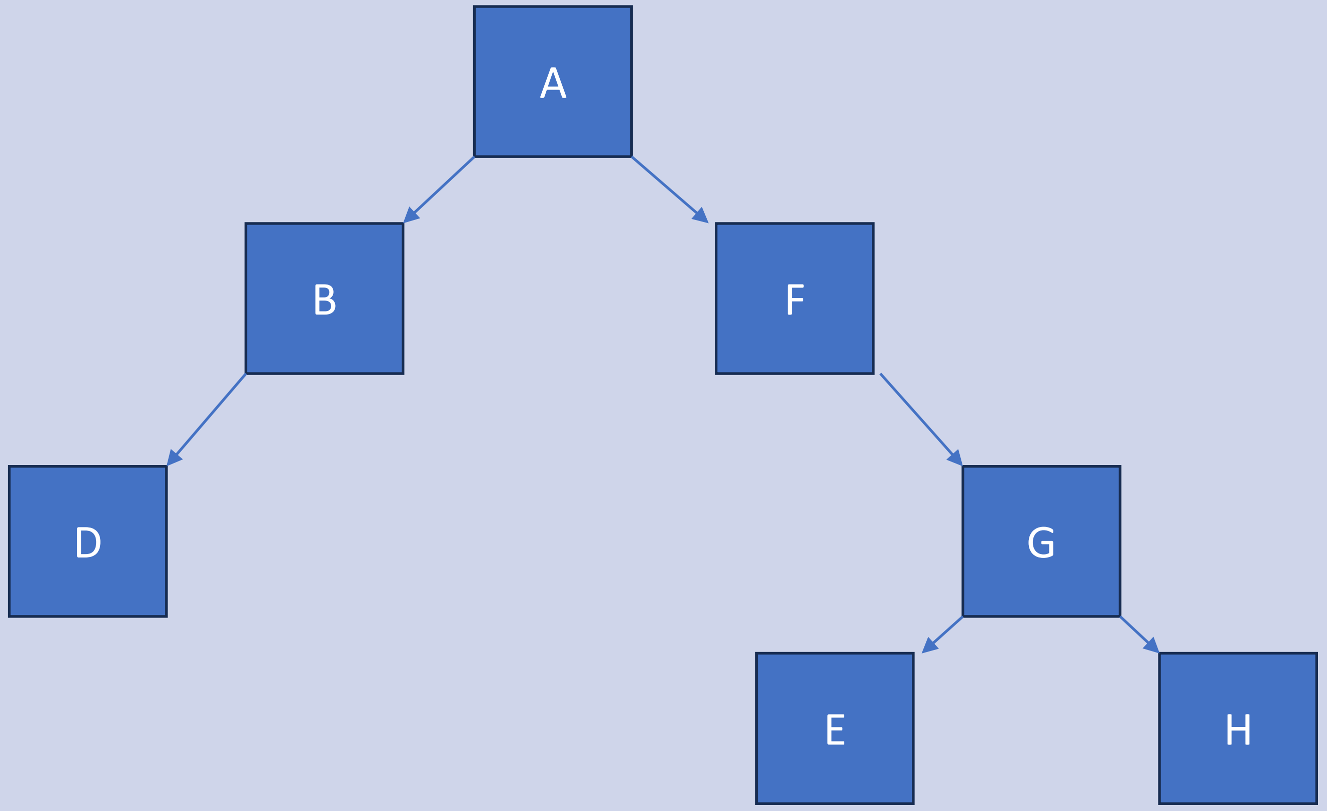
```
//base class portion that the students need not required to write in the answer

class Circle //Base class
{
        protected double radius, area;
        Circle(double r) //base class constructor to initialise the data members
        {
                radius=r;
                area=0.0;
        }
        void cal_area( )
        {
                area=3.142*radius*radius;
        }
        void display( )
        {
                System.out.println("Area="+area);
        }
}
```

```
class Volume extends Circle //extends keyword establish inheritance b/w base class & child class
{
        double height, volume;
        Volume(double r, double h)
        {
                super(r); //base class constructor invoked through child class constructor
                          //this statement must be the first line inside the constructor
                height=h; //initialising the child class data member
                volume=0.0;
        }
        double calculate( )
        {
                cal_area( );//calling base class method [you can also write super.cal_area( )]
                volume=area*height;
        }
        void display( )
        {
                super.display();
                System.out.println("Volume="+volume);
        }
}
```

Question 11

(i)	With the help of an example, briefly explain the constant factor in time complexity.
(ii)	Answer the following questions from the diagram of a Binary Tree given below:



(a)	Name the external nodes of the right sub-tree.
(b)	State the size and depth of the tree.
(c)	**Write the post-order traversal of the above tree structure.**

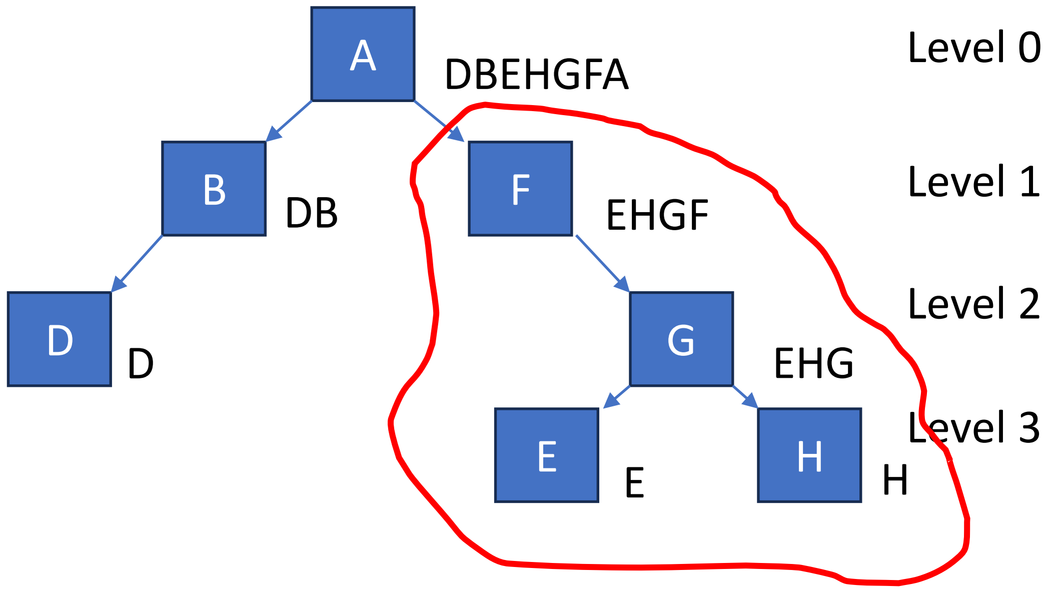
Solution:

(i)
When our algorithm is not dependent on the input size n, it is said to have a constant time complexity with order O(1). This means the run time will always be the same regardless of the input size. A good example of O(1) time is accessing a value with an array index. Other examples include push() and pop() operations on a stack where one operation adds or deletes data into/from the stack at a single point in time. Also, the constant factor plays a minimal role in the computation of complexity and hence is dropped. But when two algorithms have the same complexity, the constant factor decides which is faster.
For example, suppose algo1 requires N^2 time and algo2 req. $10*N^2+N$ time. For both the algos, the time is $O(N^2)$, but algo1 will be faster than algo2.

(ii)
(a) E, H (Leaf nodes in the right sub-tree of A, here A is the root node)

(a) Size : 7 (Total no. of nodes in the entire tree)
Depth : 4 (Total no. of nodes in the longest path or in other words, Level+1, if the level starts from 0 else only the highest level)

(a) Post-order traversal : D, B, E, H, G, F, A



Thank You

For patience watching

&

All the best for your examinations.