ISC Paper 2018
Computer Science

Maximum Marks: 70
Time allowed: 3 hours

- Candidates are allowed additional 15 minutes for only reading the paper.
- They must NOT start writing during this time.
- Answer all questions in Part I (compulsory) and six questions from Part II, choosing two questions from Section-A, two from Section-B and two from Section-C.
- All working, including rough work, should be done on the same sheet as the rest of the answer.
- The intended marks for questions or parts of questions are given in brackets.[ ].

Part – I (20 Marks)
Answer all questions.

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

Question 1.
(a) State the Commutative law and prove it with the help of a truth table. [1]
(b) Convert the following expression into its canonical POS form: [1]
F(X, Y, Z) = (X + Y) . (Y' + Z)
(c) Find the dual of [1]
(A' + B) . (1 + B') = A'+B
(d) Verify the following proposition with the help of a truth table: [1]
(P∧Q)∨(P∧~Q) = P
(e) If F(A, B, C) = A'(BC' + B'C), then find F'. [1]

Answer:
(a) Commutative law states that the interchanging of the order of operands in a Boolean equation does not change its result.
Using OR operator → A + B = B + A
Using AND operator → A * B = B * A
Truth Table for Commutative Law

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

B	A	B+A
0	0	0
0	1	1
1	0	1
1	1	1

A + B = B + A

(b) F(X, Y, Z) = (X + Y').(Y' + Z)
By De-Morgan's theorem, we have
((X + Y').(Y' + Z))' = (X + Y')' + (Y'+ Z)'
=X'.Y'' + Y''.Z'
= X'.Y + Y.Z'
= Y.(X' + Z')
Again applying De-Morgan's theorem, we have
(Y.(X' + Z'))' = (X' + Z')'

(c) In Principle of Duality;
replace (+) by (.)
replace (.) by (+)
replace 1 b 0
Taking L.H.S. = (A' + B).(1 + B')
= (A'.B) + (0.B')
=A'.B
Again applying the principle of duality, we have = A + B'

(d)

P	Q	~Q	P∧Q	P∧~q	P∧Q ∨ P∧~Q
T	T	F	T	F	T
T	F	T	F	T	T
F	T	F	F	F	F
F	F	T	T	F	F

(e) F(A, B ,C) = A'(CC'+B1C)
F' = (A'.(BC' + B'C))'
= A''+ (BC' + B'C)'
= A + (B'C''.B''C')
= A + ((B'C.BC')

Question 2.
(a) What are the Wrapper classes? Give any two examples. [2]
(b) A matrix A[m] [m] is stored in the memory with each element requiring 4 bytes of storage. If the base address at A[1] [1] is 1500 and the address of A[4][5] is 1608, determine the order of the matrix when it is stored in Column Major Wise. [2]
(c) Convert the following infix notation to postfix form: [2]
A + (B – C*(D/E) * F)
(d) Define Big 'O' notation. State the two factors which determine the complexity of an algorithm. [2]
(e) What is exceptional handling? Also, state the purpose of finally block in a try-catch statement. [2]

Answer:
(a) A Wrapper class is a class whose object wraps or contains primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

Primitive Data Type	Wrapper Class
char	Character
short	Short

(b) Base Address, B = 1500
W = 4 Bytes
i = 4, j = 5, r = 1, c = 1
Formula for Column Major Wise
A[4][5] = B + W[m(j – c) + (I – r)]
1608 = 1500 + 4[m(5 – 1) + (4 – 1)]
1608 – 1500 = 16m + 12
16m = 96

m = 6

(c)

Symbol	Stack	Postfix
A	Empty	A
+	+	A
(	(+	A
B	(+	AB
-	(+-	AB
C	(+-	ABC
*	*(+-	ABC
(	*((+-	ABC
D	*((+-	ABCD
/	*((+-/	ABCD
E	*((+-/	ABCDE
)	*(	ABCDE+-/
*	**(	ABCDE+-/
F	**(	ABCDE+-/F
)	*	ABCDE+-/F*
		ABCDE+-/F**

(d) Big 'O' notation is a particular tool for assessing algorithm efficiency and is used to describe the performance or complexity of an algorithm. While analyzing an algorithm, we mostly consider time complexity and space complexity.
The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input. Similarly, the Space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.

(e) The exception handling in java is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.
Java finally block is a block that is used to execute important code such as closing connection, stream etc.
Java finally block is always executed whether an exception is handled or not.
Java finally block follows try or catch block.

Question 3.
The following is a function of some class which checks if a positive integer is a Palindrome number by returning true or false. (A number is said to be palindrome if the reverse of the number is equal to the original number.) The function does not use the modulus (%) operator to extract digit. There are some places in the code marked by ?1?, ?2?, ?3?, ?4?, ?5? which may be replaced by a statement/expression so that the function works properly.

```
boolean PalindromeNum(int N)
{
int rev=?1?;
int num=N;
while(num>0)
{
int f=num/10;
int s= ?2?;
int digit = num-?3?;
rev= ?4? + digit;
num/= ?5?;
}
if(rev==N)
return true;
else
return false;
}
```

1. What is the statement or expression at ?1?
2. What is the statement or expression at ?2?
3. What is the statement or expression at ?3?
4. What is the statement or expression at ?4?
5. What is the statement or expression at ?5?

Answer:

1. 0
2. 0
3. 1
4. rev*10
5. 10

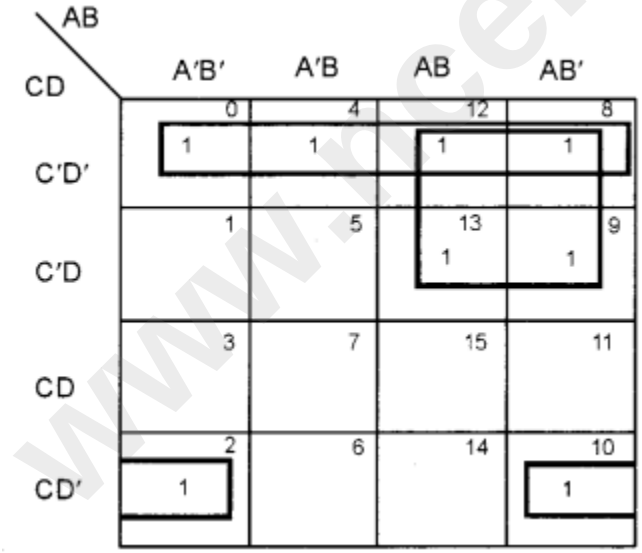
Part – II (50 Marks)

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.

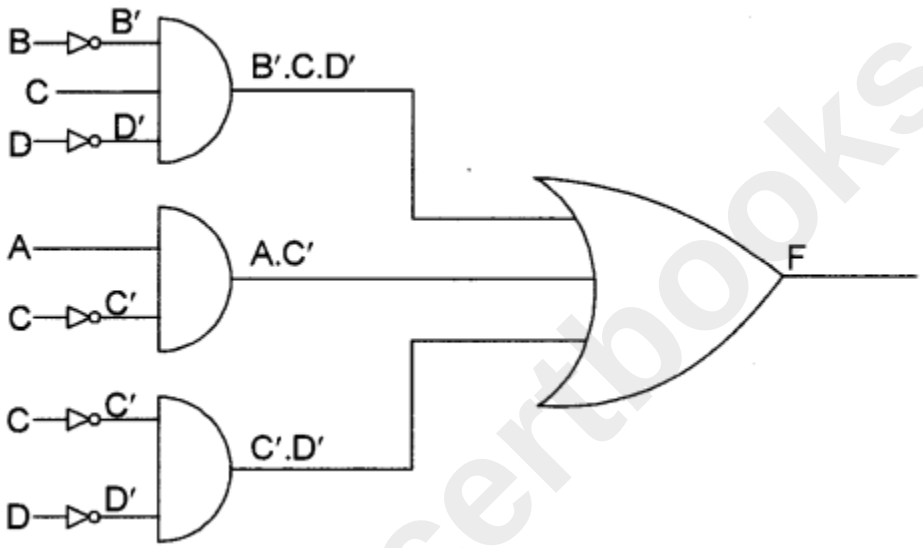
Question 4.
(a) Given the Boolean function F(A, B, C, D) = Σ (0, 2, 4, 8, 9, 10, 12, 13).
(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.eoctal, quads and pairs). [4]
(ii) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]
(b) Given the Boolean function: F(A, B, C, D) = π(3, 4, 5, 6, 7, 10, 11, 14, 15).
(i) Reduce the above expression by using the 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. Assume that the variables and their complements are available as inputs. [1]

Answer:
(a) (i) F(A, B, C, D) = Σ (0, 2, 4, 8, 9, 10, 12, 13)

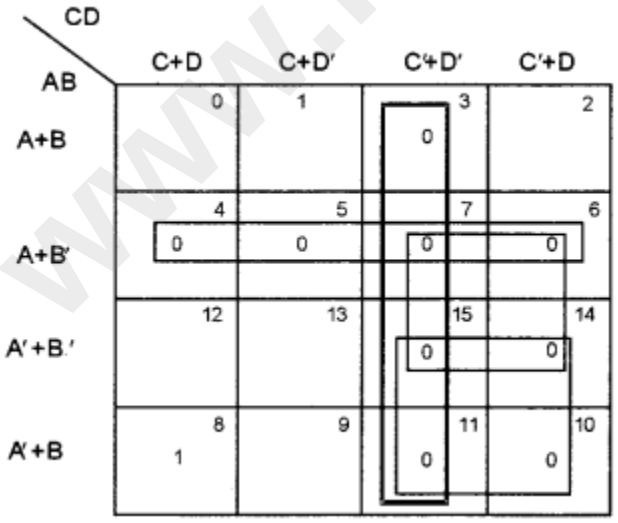


Product at cell
(0) = A'.B'C'.D'
(4) = A'.B.C'.D'
(12) = A.B.C'.D'
(8) = A.B'.C'.D' (common is C'.D')
Product at

(12) = A.B.C'.D'
(8) = A.B'.C'.D'
(9) = A.B'.C'.D
(13) = A.B.C'.D (common is A.C')
Product at
(2) = A'.B'.C.D'
(10) = A.B'.C.D' (common is B'.C.D')
Reduced expression is CD' + AC' + B'.C.D'
(ii) C'.D' + A.C' + B'.C.D'
⇒ A.C' + B'.C.D' + C'.D'
Logic Gate

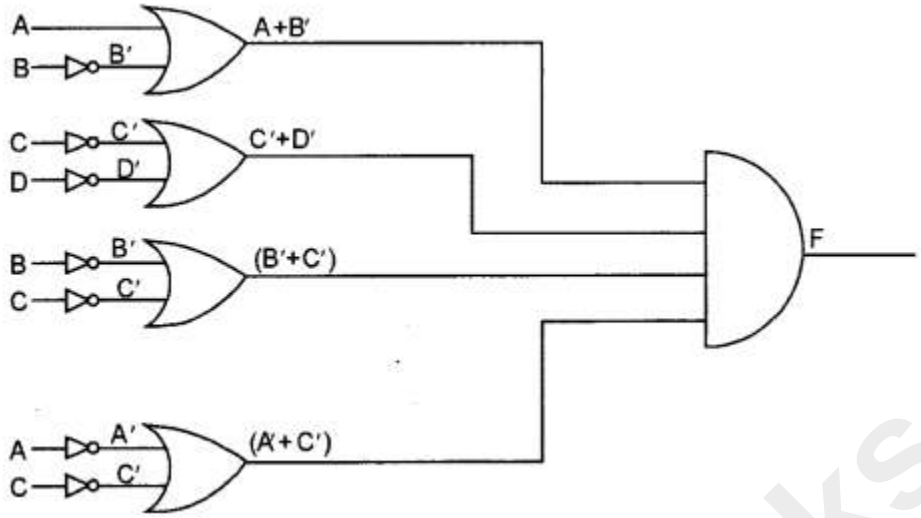


(b) (i) F(A, B, C, D) = π(3, 4, 5, 6, 7, 10, 11, 14, 15)



Reduced Expression is (A + B').(C' + D').(B' + C').(A' + C')
(ii) (A + B').(C' + D').(B' + C').(A' + C)

Logic Gate



Question 5.

(a) A training institute intends to give scholarships to its students as per the criteria given below: [5]

The student has excellent academic record but is financially weak.

OR

The student does not have an excellent academic record and belongs to a backward class.

OR

The student does not have an excellent academic record and is physically impaired.

The inputs are:

INPUTS	
A	Has excellent academic record
F	Financially sound
C	Belongs to a backward class
I	Is physically impaired

(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 SOP expression for X(A, F, C, I).

(b) Using the truth table, state whether the following proposition is a tautology, contingency or a contradiction: [3|

~(A∧B)∨(~A=>B)

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

A.(A' + B).C.(A + B)

Answer:

(a)

A	F	C	I	X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

(b)

A	B	~A	(A^B)	~(A^B)	~A=>B	~(A^B) ∨ A=>B)
0	0	1	0	1	0	0
0	0	1	0	1	0	0
0	1	1	0	1	1	1
0	1	1	0	1	1	1
1	0	0	0	1	1	1
1	0	0	0	1	1	1
1	1	0	1	0	1	0
1	1	0	1	0	1	0

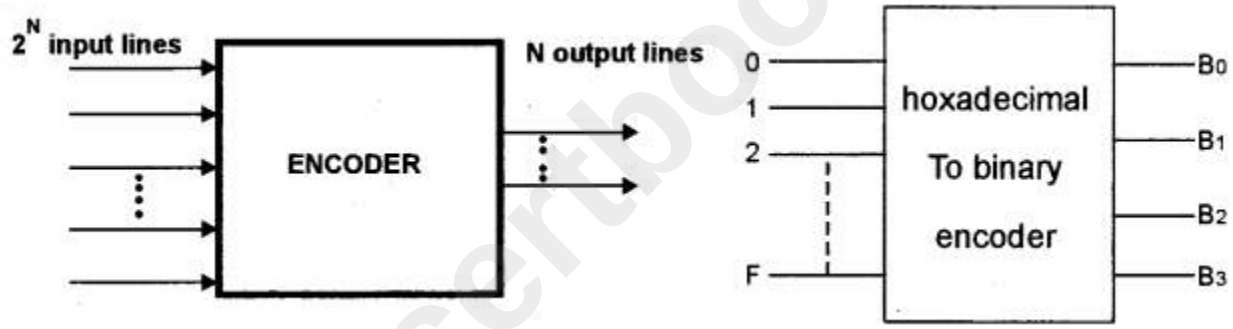
From the last column, all the values are neither all 1 ‘s nor 0’s.
It is the case of neither tautology nor contraction.
It is a contingency.

(c) A.(A’+B).C.(A + B)
= (AA’ + AB).C.(AA + AB)
= (0 + AB).C.(A + AB) (∵ AA’ = 0)
= (AB).C.(A + AB) (∵ A + AB = A)
= AB.CA

= AA.BC (∵ A.A = A)
= A.B.C

Question 6.
(a) What is an Encoder? Draw the Encoder circuit to convert A-F hexadecimal numbers to binary. State an application of a Multiplexer. [5]
(b) Differentiate between Half Adder and Full Adder. Draw the logic circuit diagram for a Full Adder. [3]
(c) Using only NAND gates, draw the logic circuit diagram for A' + B. [2]

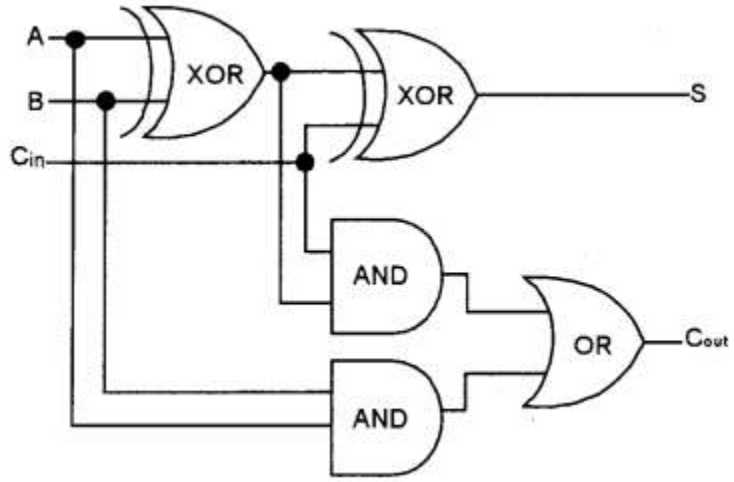
Answer:
(a) An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2An input lines and 'n' output lines, hence it encodes the information from 2An inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2An input lines with 'n' bits.



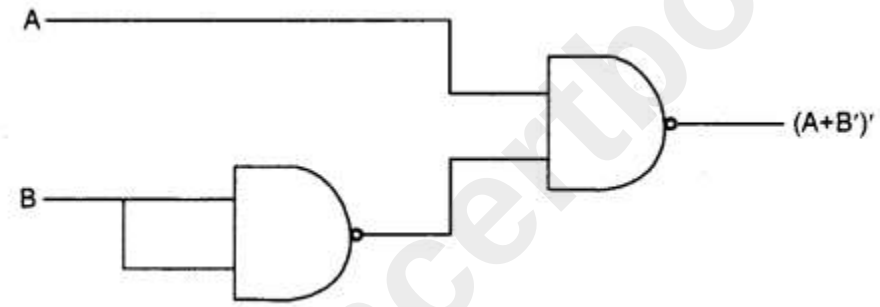
Application of a multiplexer:
The multiplexer is used to perform high-speed switching in networking.

(b) The major difference between Half Adder and Full Adder is that Half Adder adds two 1-bit numbers given as input but do not add the carry obtained from previous addition while the Full Adder, along with two 1-bit numbers can also add the carry obtained from

the previous Addition.



(c) A' + B = ((A' + B)')'
= ((A')' + B')'
= (A + B')'



Section – B
Answer any two questions.

- 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 7.
Design a class Perfect to check if a given number is a perfect number or not. [A number is said to be perfect if sum of the factors of the number excluding itself is equal to the original number]
Example: 6 = 1 + 2 + 3 (where 1, 2 and 3 are factors of 6, excluding itself) [10]
Some of the members of the class are given below:
Class name: Perfect

Data members/instance variables:
num: to store the number
Methods/Member functions:
Perfect (int nn): parameterized constructor to initialize the data member num=nn
int sum_of_factors(int i): returns the sum of the factors of the number(num), excluding itself, using a recursive technique
void check(): checks whether the given number is perfect by invoking the function sum_of_factors() and displays the result with an appropriate message
Specify the class Perfect giving details of the constructor(), int sum_of_factors(int) and void check(). Define a main() function to create an object and call the functions accordingly to enable the task.

Answer:

```
import java.io.*;
import java.util.Scanner;
class Perfect{
private int num;
private int f;
public Perfect(int num) {
this.num = num;
f = 1;
}
public int sumofFactors(int i) {
if(i==f){
return 0;
}
elseif(i%f==0)
return f++ + sumOfFactors(i);
else{
f++;
return sumOfFactors(i);
}
}
public void check() {
if(num==sumOfFactors(num))
System.out.println(nxun + " is a Perfect Number");
else
System.out.println(num + " is not a Perfect Number");
}
public static void main(String args[ ])throws IOException {
Scanner sc = new Scanner(System.in);
System.out.print("Enter the number:");
int n = sc.nextInt();
```

```
Perfect obj = new Perfect(n);
obj.check();
}
}
```

Question 8.

Two matrices are said to be equal if they have the same dimension and their corresponding elements are equal. [10]

For example, the two matrices A and B is given below are equal:

Matrix A

1	2	3
2	4	5
3	5	6

Matrix B

1	2	3
2	4	5
3	5	6

Design a class EqMat to check if two matrices are equal or not. Assume that the two matrices have the same dimension.

Some of the members of the class are given below :

Class name: EqMat

Data members/instance variables:

a[][] : to store integer elements

m: to store the number of rows

n: to store the number of columns

Member functions/methods:

EqMat: parameterized constructor to initialise the data members m = mm and n = nn

void readArray(): to enter elements in the array

int check(EqMat P, EqMat Q): checks if the parameterized objects P and Q are equal and returns 1 if true, otherwise returns 0

void print(): displays the array elements

Define the class EqMat giving details of the constructor), void readarray( ), int check(EqMat, EqMat) and void print( ). Define the main( ) function to create objects and call the functions accordingly to enable the task.

Answer:

```
import java.io.*;
import java.util.Scanner;
class EqMat{
private int a[][];
private static int m;
private static int n;
public EqMat(int mm, int nn) {
m = mm;
n = nn;
```

```
a = newint[m][n];
}
public void readArray( )throws IOException {
Scanner sc = new Scanner(System.in);
for(int i = 0; i < m; i++) {
for(int j = 0; j < n; j++) {
a[i][j] = sc.nextInt();
}
}
}
public static boolean check(EqMat p, EqMat q) {
boolean flag = true;
for(int i = 0; i < m; i++) {
for(int j = 0; j < n; j++) {
if(p.a[i][j] !=q.a[i][j])
return false;
}
}
return flag;
}
public void print() {
for(int i = 0; i < m; i++) {
for(int j = 0; j < n; j++) {
System.out.print(a[i] [j] + "\t");
}
System.out.println();
}
}
public static void main(String args[ ]) throws IOException {
Scanner sc = new Scanner(System.in);
System.out.print("Number of rows: ");
int rows = sc.nextInt();
System.out.print("Number of columns: ");
int columns = sc.nextInt();
EqMat obj 1 = new EqMat(rows, columns);
EqMat obj2 = new EqMat(rows, columns);
System.out.println("Enter elements for first matrix:");
obj1.readArray();
System.out.println("Enter elements for second matrix:");
obj2.readArray();
System. out.println("First Matrix:");
obj1.print();
System.out.println("Second Matrix:");
obj2.print();
```

```
if(check(obj1, obj2))
System.out.println("Both Matrices are Equal");
else
System.out.println("Matrices are not Equal");
}
}
```

Question 9.
A class Capital has been defined to check whether a sentence has words beginning with a capital letter or not. [10]
Some of the members of the class are given below:
Class name: Capital
Data member/instance variable:
sent: to store a sentence
freq: stores the frequency of words beginning with a capital letter
Member functions/methods:
Capital () : default constructor
void input (): to accept the sentence
boolean isCap(String w): checks and returns true if the word begins with a capital letter, otherwise returns false
void display(): displays the sentence along with the frequency of the words beginning with a capital letter
Specify the class Capital, giving the details of the constructor( ), void input( ), boolean isCap(String) and void display( ). Define the main( ) function to create an object and call the functions accordingly to enable the task.

Answer:

```
import java.io.*;
import java.util. Scanner;
import java.util.StringTokenizer;
public class Capital{
private String sent;
private int freq;
public Capital() {
sent = new String();
freq = 0;
}
public void input() throws IOException {
Scanner sc = new Scanner(System.in);
System.out.print("Enter the sentence: ");
sent = sc.next();
}
boolean isCap(String w) {
```

```
char ch = w.charAt(0);
if(Character.isLetter(ch) && Character.isUpperCase(ch))
return true;
return false;
}
public void display() {
StringTokenizer st = new StringTokenizer(sent, "" , ?!");
int count = st.countTokens();
forint i = 1; i<= count; i++) {
String word = st.nextToken();
if(isCap(word))
freq++;
}
System.out.println("Sentence: " + sent);
System.out.println("Frequency: " + freq);
}
public static void main(String args[ ])throws IOException {
Capital obj = new Capital();
obj.input();
obj.display();
```

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 10.
A superclass Number is defined to calculate the factorial of a number. Define a subclass Series to find the sum of the series S = 1! + 2! + 3! + 4! + ……. + n! [5]
The details of the members of both classes are given below:
Class name: Number
Data member/instance variable:
n: to store an integer number
Member functions/methods:
Number(int nn): parameterized constructor to initialize the data member n=nn
int factorial(int a): returns the factorial of a number
(factorial of n = 1 × 2 × 3 × …… × n)
void display()

Class name: Series
Data member/instance variable:
sum: to store the sum of the series
Member functions/methods:
Series(…) : parameterized constructor to initialize the data members of both the classes
void calsum(): calculates the sum of the given series
void display(): displays the data members of both the classes
Assume that the superclass Number has been defined. Using the concept of inheritance, specify the class Series giving the details of the constructor(…), void calsum() and void display().
The superclass, main function and algorithm need NOT be written.

Answer:

```
import java.io.*;
import java.util. Scanner;
class Number{
int n;
public Number(int nn) {
n = nn;
}
public int factorial(int a) {
if(a <= 1)
return 1;
return a * factorial(--a);
}
public void display() {
System.out.println(“Number: ” + n);
}
}
class Series extends Number {
int sum;
public Series(int n) {
super(n);
sum = 0;
}
public void calcSum() {
for(int i = 1; i <= n; i++) {
sum += super.factorial(i);
}
}
public void display() {
super, display();
System.out.println(“Series sum: ” + sum);
```

```
}
}
class Factorial{
public static void main(String args[ ]) throws IOException {
Scanner sc = new Scanner(System.in);
System.out.print("Enter the number: ");
int niun = sc.nextInt();
Series obj = new Series(num);
obj.calcSum();
obj.display();
}
}
```

Question 11.

A register is an entity which can hold a maximum of 100 names. The register enables the user to add and remove names from the topmost end only.

Define a class Register with the following details:

Class name: Register

Data members/instance variables:

stud[]: array to store the names of the students

cap: stores the maximum capacity of the array to point the index of the top end

top: to point the index of the top end

Member functions:

Register (int max) : constructor to initialize the data member cap = max, top = -1 and create the string array

void push(String n): to add names in the register at the top location if possible, otherwise display the message "OVERFLOW" String pop(): removes and returns the names from the topmost location of the register if any, else returns "$$"

void display (): displays all the names in the register

(a) Specify the class Register giving details of the functions void push(String) and String pop().

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]

Answer:

(a)
```
import java.io.*;
import java.util.Scanner;
class Register{
private String stud[];
private int cap;
private int top;
public Register(int max) {
```

```
top = -1;
cap = max;
if(cap > 100)
cap = 100;
stud = new String[cap];
}
public void push(String n) {
if(top + 1 < cap)
stud[++top] = n;
else
System.out.println(“OVERFLOW”);
}
public String pop() {
if(top == -1)
return “$$”;
else
return stud[top--];
}
public void display() {
if(top== -1)
System.out.println(“Register empty.”);
else
{
System.out.println(“StudentList:”);
for(int i = 0; i <= top; i++)
System.out.println(stud[i]);
}
}
public static void main(String args[ ]) throws IOException {
Scanner sc = new Scanner(System.in);
System. out.print(“Maximum size: ”);
int max = sc.nextInt();
Register obj = new Register(max);
while(true) {
System.out.println(“1. Push”);
System.out. println(“2. Pop”);
System.out.println(“3. Display”);
System.out.print(“Enter your choice: ”);
int choice = sc.nextInt();
switch(choice) {
case 1:
System.out.print(“Student Name: ”);
String n = sc.next();
obj.push(n);
```

```
break;
case 2:
n = obj.pop();
if(n.equals("$$"))
System.out.println("UNDERFLOW!");
else
System.out.println(n + "popped.");
break;
case 3:
obj.display();
break;
default:
System.out.println("Exiting...");
return;
}
}
}
}
```

(b) The entity used in the above data structure is stack. Stack works upon the principle of LIFO i. e., Last In First Out.

Question 12.

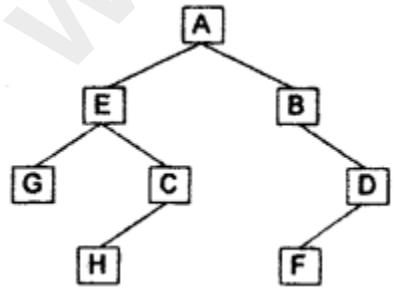
(a) A linked list is formed from the objects of the class Node. The class structure of the Node is given below: [2]

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

Write an Algorithm OR a Method to search for a number from an existing linked list. The method declaration is as follows:

void FindNode(Node str, int b)

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



(i) Write the inorder traversal of the above tree structure. [1]

(ii) State the height of the tree, if the root is at level 0 (zero). [1]

(iii) List the leaf nodes of the tree. [1]

Answer:

(a) public void FindNode(Node str, int d) {
Node str = start;
intpos= 1;
while(str != null) {
if(str.data == d) {
System.out.println(d + “found at position” + pos);
break;
}
else{
str = str.next;
pos++;
}
}
if(str==null)
System.out.println(d + “not found.”);
}
(b) (i) Inorder Traversal = E A B
= G E C A B D
= G E C H A B D F
(ii) Height of Tree = 4
(iii) Leaf nodes of the tree are those nodes that do not have any child. Therefore, leaf nodes are H and F.