**Computer Science**
**Paper 1**
**Theory**
**Year 2013**
**Part I**

Answer all questions.

While answering questions to this part, indicate briefly your working and reasoning, wherever required.

**Question 1.**

a) State the principle of duality. Write the dual of:
(P+Q').R.1=P.R+Q'.R **[ 2 ]**

b) Minimize the expression using Boolean laws:
F= (A+B').(B+CD)' **[ 2 ]**

c) Convert the following cardinal form of expression into its canonical form :
F(P,Q,R)= π(1,3) **[ 2 ]**

d) Using a truth table verify:
(~P→Q) ^P = ( P^~Q) ˅(P ^Q) **[ 2 ]**

e) If A=1 and B=0, then find :

i) (A'+1).B
ii) (A+B')' **[ 2 ]**

**Question 2.**

a) Differentiate between throw and throws with respect to exception handling. **[ 2 ]**

b) Convert the following infix notation to its postfix form:
E*(F / (G-H) * I) + J
**[ 2 ]**

c) Write the algorithm for push operation (to add elements ) in an array based stack. **[ 2 ]**

d) Name the file stream classes to :
i) Write data to a file in binary form.
ii) Read data from a file in text form. **[ 2 ]**

e) A square matrix M[][] of size 10 is stored in the memory with each element requiring 4 bytes of storage. If the base address at M[0][0] is 1840, determine the address at M[4][8] when the matrix is stored in row major wise. **[ 2 ]**

**Question 3.**

a) The following function Recur() is a part of some class. What will be the output of the function Recur() when the value of n is equal to 10. Show the dry run/working.

```
public void Recur(int n)
 {
   if(n>1)
   {
     System.out.print(n+" ");
     if(n%2!=0)
     {
       n=3*n+1;
       System.out.print(n+" ");
      }
```

```
        Recur(n/2);
      }
    }
```

b) The following function is a part of some class. Assume n is a positive integer. Answer the given questions along with dry run / working.

```
int unknown(int n)
{
   int i,k;
   if(n%2==0)
   {
      i=n/2;
      k=1;
    }
   else
   {
      k=n;
      n--;
      i=n/2;
    }
    while(i>0)
    {
       k=k*i*n;
       i--;
       n--;
    }
    return k;
}
```

**Part II**

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

**Section A**

**Answer any three questions**

**Question 4.**

a) Given the Boolean function : F(A,B,C,D ) = ∑(0,2,4,5,8,9,10,12,13)

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

ii) Draw the logic gate diagram of the reduced expression. Assume that the variables and their complements are available as inputs. **[ 1 ]**

b) Given the Boolean Function :F(P,Q,R,S) = ∏(0,1,3,5,7,8,9,10,11,14,15)

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

ii) Draw the logic gate diagram of the reduced expression. Assume that the variables and their complements are available as inputs. **[ 1 ]**

**Question 5.**

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

- If the Centre and forward players perform well but Defenders do not perform well.

  OR
- If Goal keeper and Defenders perform well but the Centre players do not perform well.

  OR
- If all the players perform well.

The inputs are:

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

(In all of 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.]

a) Draw the truth table for the inputs and outputs given above and write the POS expression for X(C,D,F,G). **[ 5 ]**

b) Reduce X(C,D,F,G) using Karnaugh's map. Draw the logic gat diagram for the reduced POS expression for X(C,D,F,G) using AND and OR gate. You may use gates with two or more inputs. Assume that the variable and their complements are available as inputs.

**[ 5 ]**

**Question 6.**

a) In the following truth table x and y are inputs and B and D are outputs: **[ 3 ]**

| **INPUT** | | **OUTPUT** | |
|---|---|---|---|
| X | Y | B | D |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

Answer the following questions:

i) Write the SOP expression for D.

ii) Write the POS expression for B.

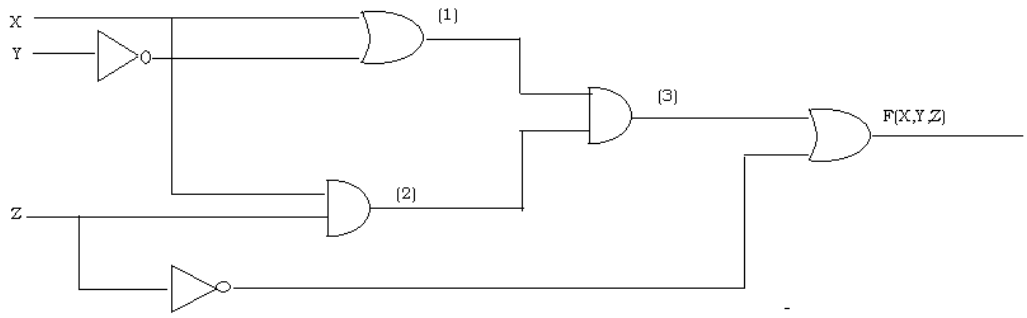
iii) Draw the logic diagram for the SOP expression derived for D, using only NAND gates.

b) Using a truth table, verify if the following proposition is valid or invalid: **[ 3 ]**

$$(A \rightarrow B) \wedge (B \rightarrow C) = (A \rightarrow C)$$

c) From the logic circuit diagram given below, name the outputs (1), (2) and (3). Finally derive the Boolean expression and simplify it to show that it represents a logic gate. Name and draw the logic gate.

**[ 4 ]**




**Question 7.**

a) What are decoders? How are they different from encoders? **[ 2 ]**

b) Draw the truth table and a logic gate diagram for a 2 to 4 decoder and briefly explain its working. **[ 4 ]**

c) A combinational logic circuit with three inputs P, Q, R produce output 1 if and only if an odd number of 0's are input.
   i) Draw its truth table.
   ii) Derive a canonical SOP expression for the above truth table.
   iii) Find the complement of the above derived expression using De Morgan's theorem and verify if it is equivalent to its POS expression.

## Section B

Answer any 2 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.

**Question 8.**

An emirp number is a number which is prime backwards and forwards. Example: 13 and 31 are both prime numbers. Thus 13 is an emirp number.

Design a class Emirp to check if a given number is Emirp number or nt. Some of the members of the class are given below:

| | | |
|---|---|---|
| **Class Name** | **:** | **Emirp** |
| **Data Members** | | |
| n | : | stores the number |
| rev | : | stores the reverse of the number |
| f | : | stores the divisor |
| **Member functions** | | |
| Emirp(int nn) | : | to assign n=nn, rev=0, and f=2 |
| int isprime(int x) recursive | : | check if the number is prime using the technique and return 1 if prime otherwise return 0. |
| void isEmirp() the | : | reverse the given number and check if both original number ad the reverse number are prime, by invoking the function isprime(int) and display the result with an appropriate message. |

Specify the class Emirp giving details of the constructor(int), int isprime(int) and void isEmirp(). Define the main function to create an object and call the methods to check for Emirp number.

**[ 10]**

**Question 9.**

Design a class Exchange to accept a sentence and interchange the first alphabet with the last alphabet for each word in the sentence, with single letter word remaining unchanged. The words in the input are separated by a single blank space and terminated by a full stop.

Example: Input: It is a warm day.
Ouput: tI si a marw yad

Some of the data members and member functions are given below:

| **Class Name** | **:** | **Exchange** |
|---|---|---|
| **Data members** | | |
| sent | : | stores the sentence |
| rev | : | to store the new sentence |
| size | : | stores the length of the sentence |
| **Member functions** | | |
| Exchange() | : | default constructor |
| void readsentence() | : | to accept the sentence |
| void exfirstlast() | : | extract each word and interchange the first and last alphabet of the word and form a new sentence. |
| void display() | : | display the original sentence along with the new changed sentence. |

Specify the class Exchange giving details of the constructor(), void readsentence(), void exfirstlast() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

**[ 10]**

**Question 10.**

A class Matrix contains a two dimensional integer array of order [ m x n ]. The maximum value possible for both m and n is 25. Design a class Matrix to find the difference of the two matrices. The details of the members of the class are given below:

| **Class name** | **:** | **Matrix** |
|---|---|---|
| **Data members** | | |
| arr[][] | : | stores the matrix element |
| m | : | integer to store the number of rows |
| n | : | integer to store the number of columns |
| **Member functions:** | | |
| Matrix(int mm, int nn) and | : | to initialize the size of the matrix m=mm<br>n=nn |
| void fillarray() | : | to enter the elements of the matrix |
| Matrix SubMat(Matrix A) of | : | subtract the current object from the matrix |

| | | |
|---|---|---|
| | | parameterized object and return the resulting object. |
| void display() | : | display the matrix elements. |

Specify the class Matrix giving details of the constructor(int,int), void fillarray(),Matrix SubMat(Matrix) and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

**[ 10]**

**Section C**

**Answer any 2 questions**.

Each program/algorithm should be written in such a way that it clearly depicts the logic of the problem step wise. This can also be achieved by using comment in the program and mnemonic names or pseudo codes for algorithms.

(Flowcharts are not required)

**The programs must be written in Java.**

**The algorithm must be written in general standard form wherever required/specified.**

**Question 11.**

A super class Perimeter has been defined to calculate the perimeter of a parallelogram. Define a sub class Area to compute the area of the parallelogram by using the required data members of the super class. The details are given below:

| | | |
|---|---|---|
| **Class name** | **:** | **Perimeter** |
| **Data Members** | | |
| a | : | to store the length in decimal |
| b | : | to store the breadth in decimal |
| | | |
| **Member functions:** | | |
| Perimeter(...) | : | parameterized constructor to assign values to data Members |
| double Calculate() | : | calculate and return the perimeter of a parallelogram as<br>2 *(length +breadth) |
| void show() | : | to display the data members along with the perimeter<br>of the parallelogram. |
| | | |
| **Class name** | **:** | **Area** |
| **Data Members** | | |
| h | : | to store the height in decimal |
| b | : | to store the area of the parallologram |
| | | |
| **Member functions:** | | |
| Area(...) | : | parameterized constructor to assign values to data Members of both classes. |
| void doarea() | : | computes the area( breadth * height). |

| | | |
|---|---|---|
| void show() | : | to display the data members of both the classes along<br>with the area and perimeter of the parallelogram. |

Specify the class Perimeter giving details of the constructor(...), double Calculate() and void show(). Using the concept of inheritance, specify the class Area giving details of he constructor(...), void doarea() and void show(). The main function and algorithm need not be written.

**[ 10 ]**

**Question 12.**

A doubly queue is a linear data structure which enables the user to add and remove integers from either ends, i.e. from front or rear. Define a class Dequeue with the following details:

| | | |
|---|---|---|
| **Class name** | **:** | **Dequeue** |
| **Data members** | | |
| arr[] | : | array to hold upto 100 integer elements |
| lim | : | stores the limit of the dequeue |
| front | : | to point to the index of front end |
| rear | : | to point to the index of rear end |
| **Member function** | | |
| Dequeue(int l) | : | constructor to initialise the data members lim=l; front=rear=0 |
| void addfront(int val) | : | to add integer from the front if possible else display the<br>message ("Overflow from front") |
| void addrear(int val) | : | to add integer from the rear if possible else display the<br>message ("Overflow from rear") |
| int popfront() | : | returns element from front, if possible otherwise returns -9999. |
| int poprear() | : | returns element from rear, if possible otherwise returns -9999. |

Specify class Dequeue giving details of the constructor(), void addfront(int), void addrear(int), int popfront(), and int poprear().
The main function and algorithm need not be written. **[ 10 ]**

**Question 13.**

a) A linked list is formed from the objects of the class,

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

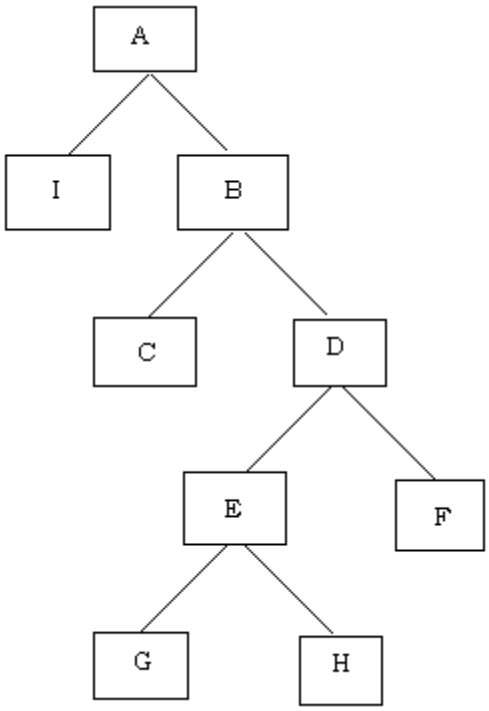
Write an algorithm or a method to count the numbr of nodes in the linked list. The method declaration is given below:
int count(Node ptr_start) **[4]**

b) What is the worst case complexity f the following code segment:

```
i)
for(int p=0; p<N;p++)
{
        for(int q=0;q<M;q++)
        {
                Sequence of statements
        }
}
for(int r=0;r<X;r++)
{
        Sequence of statements
}
```

ii) How would the complexity change if all the loops went up to the same limit N? **[2]**

c) Answer the following questions from the diagram of the binary tree given below:




i) Preorder traversal of tree
ii) Children of Node E
iii) Left subtree of Node D
iv) Height of the tree when the root of the tree is at level 0. **[4]**