Solved Paper

C.B.S.E 2009

(Class XII)

Time allowed : 3 hours   **Maximum Marks : 100**

General Instructions :

    (i)   All questions are compulsory.

    (ii) Programming Language : C++

Delhi   **91/1**

1. (a) What is the difference between call by value and call by reference ? Give an example in C++ illustrate both.   **2**

    (b) Write the names of the header files to which the following belong :   **1**

      (i) puts()

      (ii) sin()

    (c) Rewrite the following program after removing the syntactical errors (if any). Underline each correction.   **2**

```
#include [iostream.h]
#include [stdio.li]
class Employee
{
        int EmpId = 901;
        char EName[20];
public
        Employee(){}
        void Joining() {cin>>EmpId; gets (EName);}
        void List() {cout<<EmpId<<" : " <<EName<<endl;}
};
void main()
{
        Employee E;
        Joining. E ();
        E. List ()
}
```

    (d) Find the output of the following program :   **3**

```
#include<iostream.h>
void main()
{
        int X[] = {10, 25, 30, 55, 110};
        int *p = X;
        while (*p < 110)
        {
            if (*p%3 != 0)
                        *p = *p + 1;
                else
                        *p = *p + 2;
                p++;
        }
        for (int I = 4; I>=1 ; 1 I– –)
        {
                cout<<X [I] << "*" ;
                if (I%3 == 0) cout<<endl;
        }
        cout<<X [0] * 3<<endl;
}
```

(e) Find the output of the following program :   **2**

```
#include<iostream.h>
#include<ctype.h>
void Encode (char Info[], int N);
void main()
{
    char Memo[]="Justnow";
    Encode (Memo, 2);
    cout<<Memo<<endl;
}
void Encode (char Info []. int N)
{
    for (int I=0;Info [I] !='0'; I++)
    if (I%2==0)
            Info [I] = Info [I] –N;
            else if (islower(Info[I]))
            Info[I]=toupper(Info[I]);
            else
            Info[I]=Info[I]+N;
}
```

(f) Study the following program and select the possible output from it :   **2**

```
#include<iostream.h>
#include<stdlib.h>
const int LIMIT=4;
void main()
{
    randomize();
    int Points;
    Points=100+random(LIMIT);
    for (int P=Points;P>=100;P—)
            cout<<P<<"#";
    cout<<endl;
}
```

  (i) 103#102#101#100#
 (ii) 100#101#102#103#
(iii) 100#101#102#103#104#
 (iv) 104#103#102#101#100#

2. (a) What is copy constructor ? Give an example in C++ to illustrate copy constructor.   **2**
    (b) Answer the questions (i) and (ii) after going through the following class: **2**

```
class WORK
{
        int WorkId;char WorkType;
public:
        ~WORK()                    //Function 1
        {cout<<"Un-Allocated"<<endl;}
        void status()              //Function 2
        {cout<<WorkId<<" : "<<Work Type<<endl;}
        WORK()                     //Function 3
        {WorkId=10 ; WorkType='T';}
        WORK(WORK & W)             //Function 4
        {
                WorkId=W.WorkId+12; Work Type=W.WorkType+1
        }
};
```

  (i) Which member function out of Function 1, Function 2, Function 3 and Function **4**
       shown in the above definition of class **WORK** is called automatically, when the
       scope of an object gets over ? Is it known as Constructor OR Destructor OR Overloaded
       Function OR Copy Constructor ?
 (ii) WORK W;   //Statement 1
       WORK Y (W);   //Statement 2

Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class WORK will be called on execution of statement written as Statement 2 ? What is this function specifically known as out of Destructor or Copy Constructor or Default Constructor ?

(c) Define a class RESORT in C++ with following description : **4**

Private Members :
- Rno   //Data member to store Room No.
- Name   //Data member to store customer name.
- Charges   //Data member to store per day charges.
- Days   //Data member to store number of days of stay.
- COMPUTE()//A function to calculate and return Amount as Days*Charges and if the value of Day*Charges is more than 11000 then as 1.02*Days*Charges

Public Members
- Getinfo()   //A function to enter the content Rno, Name,
                 //Charges and Days
- Dispinfo()   //A function to display Rno, Name, Charges,
                 //Days and Amount (Amount to be displayed by
                 //calling function COMPUTE())

(d) Answer the questions (i) to (iv) based on the following : **4**

```
class Face To Face
{
        char Center Code [10];
public :
        void Input();
        void Output ();
};
class Online
{
        char webiste [50];
public
        void SiteIn ( );
        void SiteOut ( );
};
class Training : public FaceToFace, private Online
{
        long Tcode;
        float charge;
        int period;
public:
        void Register ();
        void Show ();
};
```

(i) Which type of Inheritance is shown in the above example ?

(ii) Write names of all the member functions accessible from Shown() function of class Training.

(iii) Write name of all the members accessible through an object of class Training.

(iv) Is the function Output() accessible inside the function SiteOut() ? Justify your answer.

3. (a) Write a function SORTPOINTS() in C++ to sort an array of structure Game in descending order Points using Bubble Sort. **3**

Note : Assume the following definition of structure Game.

```
struct Game.
{
    long PNo; //Player Number
    char PName [20];
    long Points;
};
```

Sample contant of the array (before sorting)

PNo	PName	Points
103	Ritika Kapur	3001
104	John Philip	2819
101	Razia Abbas	3451
105	Tarun Kumar	2971

Sample contant of the array (after sorting)

PNo	PName	Points
101	Razia Abbas	3451
103	Ritika Kapur	3001
105	Tarun Kumar	2971
104	John Philip	2819

(b) An array S[40][30] is stored in the memory along the column with each of the elements occupying 4 bytes, find out the base address and address of element S[20][15], if an element S[15][10] is stored at the memory location 7200. **4**

(c) Write a function QUEINS() in C++ to insert an element in a dynamically allocated Queue containing nodes of the following given structure : **4**

```
struct Node
{
      Int PId;          //Product Id
      Char Pname [20];
      NODE *Next;
};
```

(d) Define a function SWAPCOL() in C++ to swap (interchange) the first Column elements with the last column elements, for a two dimensional integer array passed as the argument of the function. **3**

Example : If the two dimensional array contains

2	1	4	9
1	3	7	7
5	8	6	3
7	2	1	2

After swapping of the content of 1st column, it should be:

9	1	4	2
7	3	7	1
3	8	6	5
2	2	1	7

(e) Convert the following infix expression to its equivalent postfix expression showing stack contents for the conversion. **2**

X – Y / (Z + U)* V

4. (a) Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task. **1**

```
#include <fstream.h>
class Stock
{
        long Ino;                 //Item Number
        char Item [20];           //Item Name
        int Qty;                  //Quantity
public:
        void Get(int);            //Function to enter the content
        void show();              //Function to display the content
        void Purchase (int Tqty)
        {
              Qty+=Tqty;
        }                         //Function to increment in Qty
        long KnowIno () {return Ino;}
};
        void Purchaseitem (long PINo, int PQty)
                          //PINo –> Ino of the item purchased
                          //PQty –> Number of item purchased
        {
              fstream File;
```

```
                    File.open ("ITEMS.DAT", ios : : binary|ios : : in|ios : : out);
                    int Pos=−1;
                    Stock S;
               while (Pos==−1 && File.read ((char*) &L, sizeof (S)))
               if (S. KnowIno()==PINO)
               {
                    S. Purchase (PQty); //To update the number of Items
                    Pos=File.tellg () −sizeof (S);
               //Line 1: To place the file pointer to the required position.
                    ——————————;
               //Line 2: To write the object S on to the binary file
                    ——————————;
               }
                    if (Pos==−1
                         cout<<"No updation done as required Ino not found..";
                    File.close();
               }
```

(b) Write a function COUNT_DO() in C++ to count the presence of a word 'do' in a text file "MEMO.TXT".   **2**

Example :

If the content of the file "MEMO.TXT" is as follows :

> I will do it, if you
> request me to do it.
> It would have been done much earlier.

The function COUNT_DO() will display the following message :

> Count of –do– in file: 2

Note : In the above example, 'do' occurring as a part of word done is not considered.

(c) Write a function in C++ to read and display the detail of all the users whose status is 'A' (i.e. Active) from a binary file "USER.DAT". Assuming the binary file "USER.DAT" is containing objects of class USER, which is defined as follows :   **3**

```
class USER
{
     int Uid;                    //User Id
     char Uname [20];            //User Name
     char Status;                // User Type : A Active I Inactive
public:
void Register();                //Function to enter the content
void show();                    //Function to display all data members
char Getstatus () {return Status;}.
};
```

5. (a) What are candidate keys in a table ? Give s suitable example of candidate keys in a table.   **2**

(b) Consider the following tables GARMENT and FABRIC. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).   **6**

Table : GARMENT

GCODE	DESCRIPTION	PRICE	FCODE	READYDATE
10023	PENCIL SKIRT	1150	F03	19–DEC–08
10001	FORMAL SHIRT	1250	F01	12–JAN–08
10012	INFORMAL SHIRT	1550	F02	06–JUN–08
10024	BABY TOP	750	F03	07–APR–07
10090	TULIP SKIRT	850	F02	31–MAR–07
10019	EVENING GOWN	850	F03	06–JUN–08
10009	INFORMAL PANT	1500	F02	20–OCT–08
10007	FORMAL PANT	1350	F01	09–MAR–08
10020	FROCK	850	F04	09–SEP–07
10089	SLACKS	750	F03	20–OCT–08

Table : FABRIC

FCODE	TYPE
F04	POLYSTER
F02	COTTON
F03	SILK
F01	TERELENE

(i) To display GCODE and DESCRIPTION of a each dress in descending order of GCODE.

(ii) To display the details of all the GARMENTs, which have READYDATE in between 08–DEC–07 and 16–JUN–08 (inclusive of both the dates).

(iii) To display the average PRICE of all the GARMENTs, which are made up of FABRIC with FCODE as F03.

(iv) To display FABRIC wise highest and lowest price of GARMENTs from DRESS table. (Display FCODE of each GARMENT along with highest and lowest price)

(v) SELECT SUM (PRICE) FROM GARMENT WHERE FCODE= 'F01';

(vi) SELECT DESCRIPTION, TYPE FROM GARMENT, FABRIC WHERE GARMENT.FCODE = FABRIC. FCODE AND GARMENT. PRICE>=1260;

(vii) SELECT MAX (FCODE) FROM FABRIC;

(viii) SELECT COUNT (DISTINCT PRICE) FROM FABRIC;

6. (a) Verify X'Y + X.Y' + X'Y' = (X' + Y') using truth table. **2**

(b) Write the equivalent Boolean Expression of the following Logic Circuit : **2**
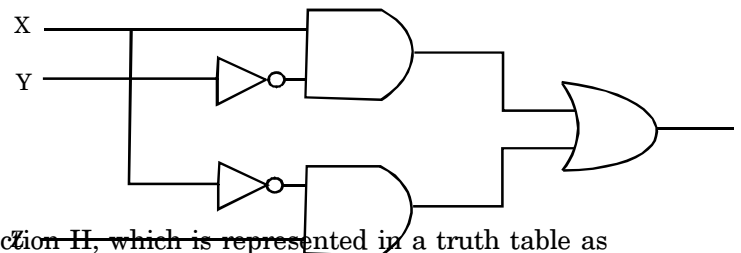
(c) Write the POS form of a Boolean function H, which is represented in a truth table as follows : **1**

A	B	C	H
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(d) Reduce the following Boolean Expression using K–Map: **3**
F(P, Q, R, S) = Σ (1, 2, 3, 5, 6, 7, 9, 11, 12, 13, 15)

7. (a) What is the difference between Star Topology and Bus Topology of network ? **1**

(b) Expand the following abbreviations : **1**
(i) GSM
(ii) CDMA

(c) What is protocol ? Which protocol is used to search information from internet using an internet browser ? **1**

(d) Name two switching techniques used to transfer data between two terminals (computers). **1**

(e) Freshminds University of India is starting its first campus in Ana Nagar of South India with its center admission office in Kolkata. The university has 3 major blocks comprising of Office Block, Science Block and Commerce Block in the 5 KM area Campus.

As a network expert, you need to suggest the network plan as per (E1) to (E4) to the authorities keeping in mind the distances and other given parameters.
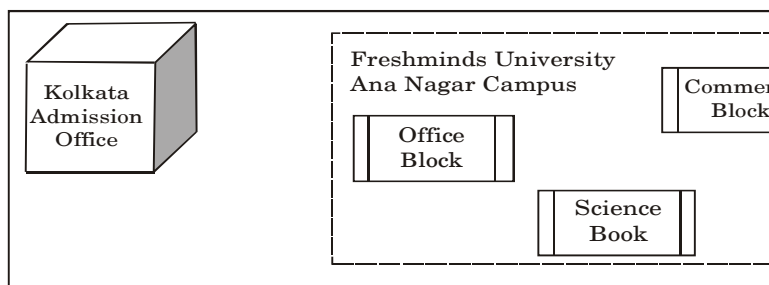
Expected Wire distances between various locations :

Office Block Science Block	90 m
Office Block to Commerce Block	80 m
Science Block Commerce Block	15 m
Kolkata Admission office to Ana Nagar Campus	2450 km

Expected number of computers to be installed at various locations in the university are as follows :

Office Block	10
Science Block	140
Commerce Block	30
Kolkata Admission office	8

(E1) Suggest the authorities, the cable layout amongst various block inside university campus for connecting the blocks. **1**

(E2) Suggest the most suitable place (i.e. block) to house the server of this university, with a suitable reason. **1**

(E3) Suggest an efficient device from the following to be installed in each of the blocks to connect all the computers : **1**
  (i) MODEM
  (ii) SWITCH
  (iii) GATEWAY

(E4) Suggest the most suitable (very high speed) service to provide data connectivity between Admission Office located in Kolkata and the campus located in Ana Nagar from the following options: **1**
  ● Telephone line
  ● Fixed–Line Dial–up connection
  ● Co-axial Cable Network
  ● GSM
  ● Leased line
  ● Satellite Connection

Outside Delhi 91

1. (a) What is the difference between Acutal Parameter and Formal Parameter ? Give an example in C++ to illustrate both types of parameters. **2**

(b) Write the names of the header files to which the following belong : **1**
  (i) **setw()**
  (ii) **sqrt()**

(c) Rewrite the following program after removing the syntactical errors (if any). Underline each correction. **2**

```
include <iostream.h>
include <stdio.h>
class MyStudent
{
      int StudentId=1001;
      char Name[20];
public
      MyStudent(){}
      void Register() {cin>>StudentId; gets (Name);}
      void Display() {cout<<StudentId<<":"<<Name<<endl;}
};
void main()
{
      MyStudent MS;
      Register. MS();
      MS. Display();
}
```

(d) Find the output of the following program :   **3**

```
#include<iostream.h>
void main()
{
      int A[] = {10, 15, 20, 25, 30};
      int *p = A;
      while (*p < 30)
      {
          if (*p%3 != 0)
              *p = *p + 2;
          else
                  *p = *p + 1;
          p++;
      }
for (int J = 0; J<=4; J++)
      {
              cout<<A [J] << "*" ;
              if (J%3 == 0) cout<<endl;
      }
      cout<<A [4] * 3<<endl;
}
```

(e) Find the output of the following program :   **2**

```
#include <iostream.h>
#include <ctype.h>
void Secret (char Msg[], int N );
void main()
{
      char SMS[]="rEPorTmE";
      Secret(SMS,2);
      cout<<SMS<<endl;
}
void Secret (chart Msg[], int N)
{
      for (int C=0;Msg [C]!='\0'; C++)
      if (C%2==0)
              Msg [C]=Msg[C]+N;
      else if (isupper (Msg[C]))
              Msg[C]=tolower(Msg[C]);
      else
              Msg[C]=Msg[C]–N;
}
```

(f) Study the following program and select the possible output from it :   **2**

```
#include <iostream.h>
```

```
#include <stdlib.h>
const int MAX=3;
void main()
{
        randomize();
        int Number;
        Number=50+random(MAX);
        for (int P=Number;P>=50;P− −)
                cout<<P<<"#";
        cout<<endl;
}
```
(i) 53#52#51#50#
(ii) 50#51#52#
(iii) 50#51#
(iv) 51#50#

2. (a) What is function overloading ? Give an example in C++ to illustrate function overloading. **2**

(b) Answer the questions (i) and (ii) after going through the following class : **2**

```
class Job
{
        int JobId; char JobType;
public:
        ~Job()                  //Function 1
        {cout<<"Resigned"<<endl;}
        Job()                   //Function 2
        { JobId=10;JobType='T';}
        void TellMe()           //Function 3
        {cout<<JobId<<":"<<JobType<<endl;}
        Job (Job &J)            //Function 4
        {
                JobId=J.JobId+10;JobType=J.JobType+1;
        }
};
```
(i) Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class Job is called automatically, when the scope of an object gets over ? Is it known as Constructor OR Destructor OR Overloaded Function OR Copy Constructor ?

(ii) Job P;   //Line 1
Job Q (P);   //Line 2
Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class Job will be called on execution of statement written as Line 2 ? What is this function specifically known as out of Destructor or Copy Constructor or Default Constructor ?

(c) Define a class HOTEL in C++ with the following description : **4**
Private Members :
● Rno   //Data member to store Room No.
● Name   //Data member to store customer name.
● Tariff   //Data member to store per day charges.
● NOD   //Data member to store number of days of stay.
● CALC()   //A function to calculate and return Amount as NOD* Tariff and if the
//value of NOD* Tariff is more than 10000 then as 1.05* NOD* Tariff.
Public Members :
● Checkin()   //A function to enter the content Rno, Name,
               //Tariff and NOD
● Checkout()   //A function to display Rno, Name, Tariff,
               //NOD and Amount (Amount to be displayed by
               //calling function CALC())

(d) Answer the questions (i) to (iv) based on the following : **4**
```
class Regular
{
```

```
                    char  School Code [10];
            public :
                    void  InRegular();
                    void  outRegular();
            };
            class Distance
            {
            char StudyCentreCode[5];
            public :
                    void Indistance ( );
                    void OutDistance ( );
            };
            class Course: public Regular, private Distance
            {
                    char Code[5];
                    float Fees ;
                    int Duration;
            public:
                    void InCourse ();
                    void Out Course ();
            };
```

  (i) Which type of Inheritance is shown in the above example ?
 (ii) Write names of all the member functions accessible from OutCourse function of class Course.
(iii) Write name of all the members accessible through an object of class Course.
(iv) Is the function InRegular() accessible inside the function InDistance() ? Justify your answer.

3. (a) Write a function SORTSCORE() in C++ to sort an array of structure. Examinee in descending order of Score using Bubble Sort. **3**

Note : Assume the following definition of structure Examinee.

```
struct Examinee
{
      long RollNo;
      char Name[20];
      float Score;
};
```

Sample Content of the array (before sorting)

RollNo	Name	Score
1001	Ravyank Kapur	300
1005	Farida Khan	289
1002	Anika Jain	345
1003	George Peter	297

Sample Content of the array (after sorting)

RollNo	Name	Score
1002	Anika Jain	345
1001	Ravyank Kapur	300
1003	George Peter	297
1005	Farida Khan	289

  (b) An array T[50][20] is stored in the memory along the column with each of the elements occupying 4 bytes. Find out the base address and address of element T[30][15], if an element T[25][10] is stored at the memory location 9800. **4**

  (c) Write a function QUEDEL() in C++ to display and delete an element from a dynamically allocated Queue containing nodes of the following given structure : **4**

```
struct NODE
{
        int Itemno;
        char Itemname[20];
        NODE *Link;
};
```

(d) Define a function SWAPARR() in C++ to swap (interchange) the first row elements with the last row elements, for a two dimensional integer array passed as the argument of the function. **3**

Example : If the two dimensional array contains

5	6	3	2
1	2	4	9
2	5	8	1
9	7	5	8

After swapping of the content of first row and last row, it should be as follows :

9	7	5	8
1	2	4	9
2	5	8	1
5	6	3	2

(e) Convert the following infix expression to its equivalent postfix expression showing stack contents for the conversion : **2**

$$A + B * (C - D) / E$$

4. (a) Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task. **1**

```
#include <fstream.h>
class Library
{
        long Ano;                   //Ano – Accession Number of the Book
        char Title[20];             //Title – Title of the Book
        int Qty;                    //Qty – Number of Books in Library
public:
        void Enter(int);            //Function to enter the content
        void Display();             //Function of display the content
        void Buy(int Tqty)
        {
                Qty+=Tqty;
        }                           //Function to increment in Qty
        long GetAno() {return Ano;}
};
void BuyBook (long BANo, int BQty)
                                    //BANo →Ano of the book purchased
                                    //BQty →Number of books purchased
{
        fstream File;
        File. open ("STOCK.DAT", ios: : binary|ios: : in|ios: : out);
        int Position=–1;
        Liberary L;
        while (Position = = –1 && File. read ((char*) &L, sizeof (L)))
           if (L. GetAno() = =BANo)
           {
           L. Buy (BQty); //To update the number of Books
           Positions=File. tellg()–sizeof (L);
        //Line 1: To place the file pointer to the required position.
           _____;
        //Line 2: To write the object L on to the binary file
           _____;
        }
           if (Position==–1)
                   cout<<"No updation done as required Ano not found...";
```

```
                        File. Close();
                    }
```

(b) Write a function COUNT_TO() in C++ to count the presence of a word 'to' in a text file "NOTES.TXT". **2**

Example :

If the content of the file "NOTES. TXT" is as follows :

It is very important to know that smoking is injurious to health. Let us take initiative to stop it.

The function COUNT_TO() will display the following message :

Count of –to– in file: 3

Note—In the above example, 'to' occurring as a part of word stop is not considered.

(c) Write a function in C++ to read and display the detail of all the members whose membership type is 'L' or 'M' from a binary file "CLUB.DAT". Assume the binary file "CLUB.DAT" contains objects of class CLUB, which is defined as follows : **3**

```
class CLUB
{
    int Mno;          //Member Number
    char Mname[20];   //Member Name
    char Type;        //Member Type: L Life Member M Monthly Member G Guest
public:
    void Register();  //Function to enter the content
    void Display();   //Function to display all data members
    char WhatType() {return Type;}
};
```

5. (a) What is the purpose of a key in table ? Give an example of a key in a table. **2**

(b) Consider the following tables DRESS and MATERIAL. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii). **6**

Table : DRESS

DCODE	DESCRIPTION	PRICE	MCODE	LAUNCHDATE
10001	FORMAL SHIRT	1250	M001	12–JAN–08
10020	FROCK	750	M004	09–SEP–07
10012	INFORMAL SHIRT	1450	M002	06–JUN–08
10019	EVENING GOWN	850	M003	06–JUN–08
10090	TULIP SKIRT	850	M002	31–MAR–07
10023	PENCIL SKIRT	1250	M003	19–DEC–08
10089	SLACKS	850	M003	20–OCT–08
10007	FORMAL PANT	1450	M001	09–MAR–08
10009	INFORMAL PANT	1400	M002	20–OCT–08
10024	BABY TOP	650	M003	07–APR–08

Table : MATERIAL

MCODE	TYPE
M001	TERELENE
M002	COTTON
M004	POLYESTER
M003	SILK

(i) To display DCODE and DESCRIPTION of a each dress in ascending order of DCODE.

(ii) To display the details of all the dresses which have LAUNCHDATE in between 05–DEC–07 and 20–JUN–08 (inclusive of both the dates).

(iii) To display the average PRICE of all the dresses which are made up of material with MCODE as M003.

(vi) To display material wise highest and lowest price of dresses from DRESS table. (Display MCODE of each dress along with highest and lowest price)

(v) SELECT SUM (PRICE) FROM DRESS WHERE MCODE='M001';

(vi) SELECT DESCRIPTION, TYPE FROM DRESS, MATERIAL WHERE DRESS. DCODE=MATERIAL. MCODE AND DRESS. PRICE>=1250;

(vii) SELECT MAX(MCODE) FROM MATERIAL;

(viii) SELECT COUNT(DISTINCT PRICE) FROM DRESS;

6. (a) State and verify absorption law using truth table.　　　　　　　　**2**

(b) Write the equivalent Boolean Expression of the following Logic Circuit :　　**2**
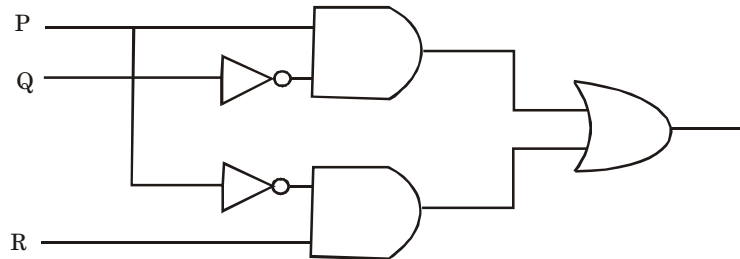
(c) Write the POS form of a Boolean function G, which is represented in a truth table as follows :　　**1**

U	V	W	G
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

(d) Reduce the following Boolean Expression using K–Map:　　**3**

H(U, V, W, Z)=Σ(0, 1, 4, 5, 6, 7, 11, 12, 13, 14, 15)

7. (a) What is the difference between LAN and WAN ?　　　　　　　　**1**

(b) Expand the following abbreviations :　　　　　　　　　　　　**1**

(i) HTTP

(ii) ARPANET

(c) What is protocol ? Which protocol is used to copy a file from/to a remotely located server ?　　**1**

(d) Name two switching techniques used to transfer data between two terminals (computers).　　**1**

(e) Eduminds University of India is starting its first campus in a small town Parampur of Central India with its center admission office in Delhi. The university has 3 major buildings comprising of Admin Building, Academic Building and Research Building in the 5 KM area Campus. As a network expert, you need to suggest the network plan as per (E1) to (E4) to the authorities keeping in mind the distances and other given parameters.
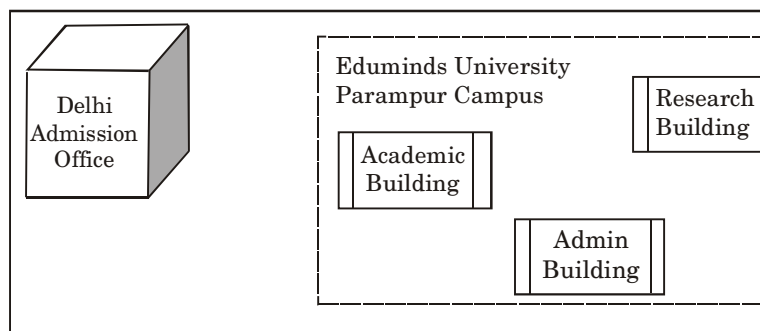
Expected Wire distances between various locations :

Research Building to Admin Building	90m
Research Building to Academic Building	80m
Academic Building to Admin Building	15m
Delhi Admission Office to Parampur Campus	1450km

Expected number of computers to be installed at various locations in the university are as follows :

Research Building	20
Academic Building	150
Admin Building	35
Delhi Admission Office	5

(E1) Suggest to the authorities, the cable layout amongst various buildings inside the university campus for connecting the buildings. **1**

(E2) Suggest the most suitable place (i.e., building) to house the server of this organisation, with a suitable reason. **1**

(E3) Suggest an efficient device from the following to be installed in each of the building to connect all the computers : **1**
  (i) GATEWAY
  (ii) MODEM
  (iii) SWITCH

(E4) Suggest the most suitable (very high speed) service to provide data connectivity between Admission Building located in Delhi and the campus located in Parampur from the following options : **1**
  ● Telephone line
  ● Fixed–Line Dial–up connection
  ● Co–axial Cable Network
  ● GSM
  ● Leased line
  ● Satellite Connection

SOLUTIONS

Delhi Set-I   **91/1**

1. **(a)** When the arguments of a function are passed by value, a copy of the value (contained in the argument) is sent from the calling routine to the function, rather than allowing the function direct access to the memory location of the variable. So, in call by value the function uses a copy of the argument value only. It cannot access the actual memory location of the variable and therefore cannot change the value of the actual argument of the calling function.

In call by reference, the function is allowed access to the actual memory location of the argument and there can be change in the value of arguments of the calling routine.

Examples of call by value and call by reference are given below:
Call by value :

```
#include<iostream.h>
#include<conio.h>
void swap (int, int);
void main ()
{
    int x,y;
    clrscr()
    cont<<"Enter the value of x and y"
    cin>>x>>y;
    cout<<"Value of x and y before passing arguments
    is"<<x<<"and"<<yendl;
    swap(x,y);
    cout<<"Value of x and y after passing argument in
    main() is"<<x<<"and"<<y;
}
void swap() int x, int y)
{
    int temp;
    temp=x;
    x=y
    y=temp;
    cout<<"Value of x and y after passing arguments
    in swap()are"<<x<<"and"<<y;
}
```

Call by reference:

```
#include<iostream.h>
#include<conio.h>
void swap(int *, int*);
void main()
{
    int x,y;
    clrscr();
    cout<<"Enter the value of x and y";
    cin>>x>>y;
    cout<<"Value of x and y before
    passing arguments
            is"<<x<<"and"<<y;
    swap(&x, &y);
    cout<<"value of x and y after
    passing arguments in
    in main()is"<<x<<"and"y;
}
void swap(int *x, int*y)
{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
    cout<<"Value of x and y after
    passing arguments
    in swap() are"<<*x<<"and"<<*y;
}
```

(b) (i) stdio.h

(ii) math.h

(c)
```
#include<iostream.h>
    #include<conio.h>
Class Employee
{
    int EmpId=901;
    char EName[20];
Public :
    Employee(){}
    void Joining () {cin>>EmpId;gets
(EName);}
    void list () {cout<<EmPId<<" :
"<<EName<<endl;}
};
    void main()
{
    Employee E;
    E. Joining ();

    E. list ();
}
```

(d) Output: 110*56*
                32*26*33

(e) output: HUqTlOu

(f) (i) 103#102#101#100#

2. (a) A copy constructor is a special constructor of function that copies the contents of an object to another, *i.e.*, constructs an object using the data of another object of the same class. **For example,**
```
#include<iostream.h>
```

```
#include<steing.h>
#include<conio.h>
class Employee
{
private:          //Access specifier
    int empcode;
    char empname [20];
    char empdesing [15];
    fleat empsalary;
    fleat da, hra, gross;
public:
    //Default constructor
    Employee (int e_code, char e_name
[25],
        char e_desig [15], float e_salary);
    //Member function prototype
declaration
    void cal_da (void);
    void col_hra (void);
    void cal_gross (void)
    void display();
    };
//Body of parameterized constructor
Employee.
Employee : : Employee (int e_code,
char_name [25], char e_desig [15] fleat
e_salary)
{
    empcode = e_code
        //Initialize object date members
    strcpy (empname, e_name);
    strcpy(empdesig, e_desig);
    empsalary = e_salary;
}
void Employee :: cal-da (void)
}
    da = 1.2*empsalary;
    cout<<"\„ Dearness      allowance:
"<<1.2*empsalary;
}
void Employee :: cal_hra (void)
{
    hra = 0.15*empsalary;
    cout<<"\„ House rent allowance: "<<hra;
}
void Employee: cal_gross (void)
{
    gross = empsalary + hra + da;
    cout <<"\„ Gross salary:<<gross;
}
void  Employee :: display ()
{
    cout<< "\„ Employee code: "<<empcode;
    cout<<  "\„ Employee    name:
    "<<empname;
    cout<< "\„ Employee designation:
    "<<empdesig;
    cout<<   "\„  Employee   salary;
    "<<empsalary;
}
```

```
void main()
{
User();
    Employee emp (10, "Miss Mini",
    lecturer, 25000);
    emp. display();
    emp. cal_da();
    emp. cal_hra();
    emp. cal_gross();
}
```

(b) (i) Function of class WORK is called automatically, when the scope of an object gets over.

It is known as Destructor.

(ii) Function 4 of class WORK will be called on execution of statement written as statement 2 It is specifically known as copy constructor.

(c) The Class is :

```
class RESORT
{
    int Rno;
    Char Name [20];
    float charges;
    int Days;
    void COMPUTE();
public :
    void Getinfo();
    void Dispinfo();
};
void RESORT :: COMPUTE ()
{
    float Amount;
    if (Days*Charges >11000)
    Amount = 1.02*Days* Charges;
    count<<"\„Amount":" << Amount;
}
void RESORT :: Getinfo ()
{
    cout<<"Enter Room No.";
    cin >>Rno;
    cout<< "\„ Enter Name:";
    cin>>Name;
    cout<<"\„ Enter charges:";
    cin>>charges;
    cout<<"\„ Enter Days :";
    cin>>Days;
}
void RESORT :: Dispinto();
{
    cout<<"\„ Room No; "<<Rno;
    cout<<"\„ Name :"<<Name;
    cout<<"\„ Charges ; "<<Charges;
    cout<<"\„ Days :"<<Days;
    COMPUTE();
}
```

(d) (i) Multiple Inheritance

(ii) Input()
Out put()
SiteIn ()

Site Out()
Register()
Tcode
Charge
Period

(iii) None

(iv) No, because both classes are different.

3. (a)
```
void SORTPOINTS (char PName [20],
long PNo, long Points)
{
    int i= 0, q=o, int temp;
    while (i<(n−1))
    {
        q=o;
        while (q<((n−p)−1))
        {if l list [q]> list [q + 1])
                { temp = list [q+1];
                list [q+1]= list [q];
                list [q] = temp;
                }
        q++;
        }
    p++;
    }
}
```

(b) For column wise allocation :

Address of s [I] [J] = Base + ((J−1)* M+(I−1)) *A

Here, M = 40

N = 30

A = 4

So, Address of S[15][10]

= Base + ((10−1)*40+(15−1))*4

7200 = Base + (360+14)*4

= Base + 354*4

Base = 7200−1416

= 5748

Now, Address of S [20][15]

= Base + ((J−1)*M + (I−1))*4

= 5784 + ((15−1)*40 + (20−1))* 4

= 5784 + ((14−40) + 19)* 4

= 5784 + (560−14)* 4

= 5784 + 574*4

= 5784 + 2296

= 8080

(c)
```
/* Function in C++ to insert in a
dynamically allocated queue */
struct NODE
{
    int PId;      //Product Id
    Char Pname[20];
    NODE * Next;
};
void QUEINS (struct NODE*start,char
data[20])
{
    struct NODE *q, * temp;
// Dynamic memory is been allocated for
a node.
    tmp = (struct NODE*) malloc (sizeof
```

```
            (struct NODE));
        tmp→Pname [20] = data [20]
        tmp→Next = NULL;
        if (start == NULL)//if list is empty
            start= tmp;
    else
    {
        //element inserted at end
        q = start;
        While (q→Next! = NULL)
            q = q→Next;
            q→Next = tmp;
        }
    }           //End of function.
```

(d)
```
#include<iostream.h>
#include<conio.h>
void SWAPCOL (int A [][], int M, int N)
{
    int i, j, temp
    for (i=o; i < M; i++)
        {
        for (j=o; j < N; j++)
        {
    temp [i][o] = A [i][o];
    A [i][o] = A[i][M–1];
    A [i][M–1] = temp [i][o];
        }
    }
}
```

(e)

Scanned Elements	Stock	Express-ion Y
(	(	$\overline{X}$
X	(	X
	(–	XY
/	(–	XY
(	(–/	XY
Z	(–/(	XYZ
+	(–/(	XYZ
U	(–/(+	XYZU
)	(–/(+	XYZU+
*	(–/	XYZU+
V	(–/*	XYZU+V
)	(/*	XYZU+V*/–

Thus, the postfix expression for X–Y/(Z+U)*V

is: X Y Z U + V * / –

4. (a) File. seekg (pos * sizeof (item)); // statement

File write ((char*) &L, size of<<)); // statement2

(b) // The function is :
```
void COUNT_DO ()
{
    fstream f;
    Cleser();
    f.open ("MEMO.TXT", ios :: in);
    char are [80];
    char ch;
    int i = o, sum = o n = o ;
While (f)
```

```
{
    f. get (ch);
    are [i] = ch;
    i++;
    if (strepy (ch, do))
{
    i– –;
    sum = sum + i;
    n++;
    }
}
cout<<"The total no. of the do is:"<<n;
}
```

(c)
```
void fune()
{
    fstream FILE;
    File. open ("USER.DAT") ios ::
                binary1 ios :: in);
    USER U:
    while (FILE. read ((char*)&U,
                sizeof (U)))
    if (U. Getstatus ()== 'A')
    U. show()
        FILE. close ();
}
```

5. (a) Candidate key: The attribute (column) or set of attributes (columns) which can identify a tuple/row uniqvely are known as candidate key is).

Also, all the key attributes of a relation that can be served as a primary key are called candidate keys.

For example, the following table Teacher, with both unique key TNo and unique TName for each teacher. In this care, the TName and subject fields are both candidate keys.

Table : Teacher

TNo	TName	Subject
T01	Mini	Computer science
T02	Prince	Maths
T03	Raj	English
T04	Vish	Economics

(b)(i) SELECTGCODE, DESCRIPTION FROM GARMENT ORDER BY GCODE DESC;

(ii) SELECT * FROM GARMENT WHERE READYMADE
            BETWEEN (08–DEC–07, 16–JUN–08);

(iii) SELECT AVG(PRINCE) FROM GARMENTG, FABRIC F WHERE G. FCODE= F.CODE
                AND
            F. FCODE = FO3;

(iv) SELECT F. FCODE, GPRICE FROM GARMENT G, FABRIC F GROUP BY F. FCODE ORDER BY G. PRICE DESC;

(v) 1600

(vi) **DESCRIPTION   TYPE**
INFORMAL SHIRT COTTON
INFORMAL PANT   COTTON
FORMAL PANT   POLYSTER

(vii) MAX () selects Numeric value not alphanumeric value. So, its error.

(viii) 7

6. (a) X´ Y + X.Y´ + X´.Y´ = (X´ + Y´)

X	Y	X'	Y'	X'.Y	X.Y'	X'.Y'	X'Y+XY'+X'.Y'	X'+Y'
0	0	1	1	0	0	1	1	1
0	1	1	0	1	0	0	1	1
1	0	0	1	0	1	0	1	1
1	1	0	0	0	0	0	0	0

∴LHS = RHS

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

(c) H (A, B, C) = (A' + B' + C')(A + B' + C')(A + B + C')

(d) F (P, Q, R, S) = S(1, 2, 3, 5, 6, 7, 9, 11, 12, 13, 15)

F(P, Q, R, S) = S + P'R + PQR'

7. (a) In BUS topology, the communication speed varies with the placing of node, but is STAR topology, every node can send and receive information with same speed.

(b) (i) **GSM** : Global System for Mobile communication

(ii) **CDMA** : Multiple Access code division

(c) A protocol is format set of rules governing data format, timing, sequencing, access control and error control required to initiate and maintain communication. The sender and receiver must use the same protocol either at an interface or end to end across a network.
IMAP is used to search information from internet using an internet below.

(d) (i) Circuit switching

(ii) Packed switching

(e) (E1) The cable layout amongst various blocks inside university campus for connecting the blocks is fibre optical cable (FOC).

(E2) The most suitable place (*i.e.* block) to house the server of this university would be science block, as this block contains the maximum number of computers, thus descreasing the cabling cost for most of the computers as well as increasing the efficiency of the maximum computers in the network.

(E3) (ii) SWITCH is to be installed in each of the blocks to connect all the computers.

(E4) Co−axial cable Network is suitable (very highly speed) service to provide data connectivity between Admission office Located in Kolkata and the campus located in Ana Nagar.

Outside Delhi Set-I   91

1. (a) The values which have been passed at the time of calling a function are called "actual parameter", whereas the variables which catch these values (in called function) are called "formal parameters".

Any number of parameters can be passed to a function being called.

Note—Actual/formal parameter are also known as Actual/formal arguments.

E.g.—

/* sending and receiving values between functions */

```
# include <iostream.h>
int calsum (int x, int y, int z)
vaid main ()
{
    int a, b, c, sum;
    cout <<"Enter the value of a, b, c";
    cin >> a >> b >> c;
    sum = calsum (a, b, c); // actual parameter
    cout << sum ;
}
int calsum (int x, int y, int z) // formal parameter
{
    int d;
    d = x + y + z;
    return (d);
}
```

The variables a, b, c, are called "actual parameter" and the variables x, y, z are called formal parameters. Here at the time of calling calsum (), the values of a, b, c will be store in x, y, and z respectively.

(b)

Function	**Header File**
setw ()	iomanip.h
sqrt ()	math.h

(c) Incorrect Program

```
include <iostream.h>#missing#sign
include <stdio.h>     #missiong#sign
class MyStudent
{
int   Student Id = 1001;
                    // variable is not
                                initialize here
char      Name [20];
public              //missing : sign
    My Student () {}
    void Register () { cin>> StudentId;
                            gets (Name);}
    void Display (){cout<<StudentId<<" :
                    "<<Name<<endl;}
}
void main ()
{
    MyStudent MS;
    Register.MS ();
                    //function name comes
                        after object name
    MS. Display ();
}
```

Correct Program :

```
#include<iostream.h>
#include<stdio.h>
class MyStudent
    {
        int      Student;
        char     Name [20];
    public :
        MyStudent () {}
        void Register () { cin>>
            StudentId; gets (Name); }
        void Display () {cout<<
    StudentId<<" : " <<Name<<endl;}
    };
    void main ()
    {
        MyStudent MS;
        MS. Register ();
        MS. Display ();
    }
```

(d)
```
void main ()
    {
        int A[] = { 10, 15, 20, 25, 30};
        int *p=A;
        while (*p<30)
        {
            if(*p%3!=0)
                *p= *p+2;
            else
                *p= *p+1;
            p++;
        }
        for(int J=0: J<=4;J++)
        {
            cout<<a[J]<<"*";
            if(J%3==0)
                cout<<endl;
        }
        cout<<A[4]*3<<endl;
    }
```
Output:
   12*
   16 * 22 * 27 *
   30 * 90

(e) Output–teRnttoe

(f) Output–52#51#50# (none of these)

2. (a) Function overloading :
One way that C++ achieves polymorphism is through the use of function overloading. In C++, two or more functions can have the same name as long as their parameter declaration are different. In this situation, the functions that have same name are said to be overloaded and the process is referred as function overloading.

//Program for demonstrate function overloading.
```
#include<iostream.h>
void cal ()
    {
int x=5, y=6;
int z=x+y;
cout<<"\n\nAddition="<<z;
    }
void cal(int x)
    {
int z=x*x;
cout<<"\n\n Square ="<<z;
    }
void cal(int x, int y)
    {
int z=x*y;
cout<<"\n\n Multiply ="<<z;
    }
void main()
    {
    cal();
    cal(12);
    cal(14,7)
    }
```
Output
   Addition = 11
   Square   = 144
   Multiply = 98

(b) (i) When the scope of an object gets over **Function–1** called automatically and this function is a **Destructor**.

(ii) On execution of Line–2 (*i.e.* **Job Q (P)**) **Function–4** will becalled and it is a **Copy Constructor.**

(c)
```
#include<iostream.h>
#include<stdio.h>
class HOTEL
{
     int Rno, NOD;
     float tariff;
     char Name[50];
float CALC()
{
float Amount;
Amount=NOD*Tariff;
if(amount>10000)
     {
     Amount=1.05*Amount;
     }
return(Amount);
}
void Checkin()
     {
     cout<<"\n Enter the Rno.=";
     cin>>Rno;
     cout<<"\n Enter Name=";
     gets(Name);
     cout<<"\n Enter tariff=";
     cin>>Tarif;
     cout<<"\n Enter the No. of
Days=";
     cin>>NOD;
     }
     void Checkout()
     {
     float Amount;
     Amount=CALC();
     cout<<"\n Room No.="<<Rno;
     cout<<"\n Name="<<Name;
     cout<<"\n Tariff="<<Tariff;
     cout<<"\n No. Of Days="<<NOD;
     cout<<"\n Amount="<<Amount;
     }
//end of function checkout
}
//end of class
```

(d) (i) Multiple Inheritance
  (ii) InRegular(), OutRegular(), InDistance(),Out Distance(), InCourse().
  (iii) In Regular (), Out R egular(), In Distance (), OutDistance (), In Course () and Out Course().
  (iv) Yes, by the Regular class object created in **"InDistance()."**

3. (a)
```
void Sortscore (struct Examinee ex[],
int n)
     {
     int i, j;
     struct Examinee t;
     for (i = 1; i<n;i++)
          {
          for (j = 0; j<n–i; j++)
               {
               if(ex[j + 1]. RollNo > ex [j]
```

RollNo)
```
          {
          t=ex[j];
          ex[j]=ex[j+l];
               ex[j+l]=t;
          }
     }
  }
}
```

(b) Given array is a Column–major order :
Hence,
$A[i, j] = Base + w [(i–L_1) + m (j–L_2)]$
where, L_1 and L_2 are lower bound of row and lower bound of column respectively.

$$T[25][10] = x + 4[25 + 50 * 10]$$
$$9800 = x + 2100$$
$$x = 9800–2100$$
$$x = 7700$$

Hence, Base Address = 7700
Now, For address of element T[30][15]
$$T[30][15] = 7700 + 4[30 + 50 \times 15]$$
$$= 7700 + 4*780$$
$$= 7700 + 3120$$
$$= 10820$$

Ans—Address of T[30][15] is 10820.

(c)
```
struct NODE
     {
     int       ItemNo;
     char      Itemname [20];
     NODE      * Link;
     };
     struct NODE, *front, *rear;
     void  Quedel()
          {
          struct NODE *t;
          if (front == NULL)
          {
          cout<<"\n   under
flow";
          }
          else
          {
          t = front;
     cout<<"\n Delete value
="<< t→ItemNo<< t→Itemname;
          if (front == rear)
               {
               front=    rear=
NULL;
               }
          else
               {
               front = front →
                         link
               }
          free(t);
```

```
                    }
                }
(d) SWAPARR (int a[][], int r, int c)
            {
            for (i = 0; i<c; i ++)
                {
                t = a [0][i];
                a[0][i] = a [r–1][i];
                a [r–1][i] = t;
                }
            }
```

(e)

Symbol Scanned	Stack	Expression
(1) A	(	A
(2) +	(+	A
(3) B	(+	AB
(4) *	(+*	AB
(5) (	(+*(	AB
(6) C	(+*(	ABC
(7) –	(+*(–	ABCD
(8) D	(+*(–	ABCD–
(9))	(+*	ABCD–*
(10) /	(+/	ABCD–*E
(11) E	(+/	ABCD–*E/+
(12))		

Ans. ABCD–*E/+

4. (a)
```
#include<fstream.h>
class library
{
        long Ano;
        char Title[20];
        int Qty;
public :
        void Enter(int);
        void Display();
        void Buy(int Tqty)
            {
            Qty + = Tqty;
            }
        long GetAno() { return Ano;}
};
void BuyBook (long BANo, int BQty)
{
fstream File;
File. Open ("STOCK. DAT",ios ::
binary | ios :: in | ios :: out);
int position=–1;
Library L;
while (position = = –1 && File. read
((char *)& L, sizeof(L)))
if (L.GetAno()== BANo)
        {
L. Buy (BQty);
position = File. tellg ()–sizeof (L);
File. seekp (position, ios :: beg);
                            // Line–1
File. write ((char *) & L, sizeof (L));
                            // Line–2
        }
```

```
if (position == –1)
cout<<"No updation done as required
Ano not found....";
File. close ();
}
```

(b)
```
#include<string.h>
#include<iostream.h>
void count_to()
    {
    char str[2];
    into count = 0;
    ifstream file ("NOTES. TXT");
    while (file. read ((char*) & str,
                        sizeof (str)))
    {
    if (strcmp (str, "to"));
        {
        count ++;
        }
    }
cout<<"\n count of _ to _ in file = "
                    << count;
}
```

(c)
```
void read_display()
{
CLUB C;
ifstream File ("CLUB. DAT");
    while (File. read ((char *) & C,
sizeof (C)))
    {
    if (C. whatType() == 'L' || C.
whatType () == 'M')
        {
        C. display ();
        }
    }
    file. close ();
}
```

5. (a) Purpose of a key
Enforcing data integrity ensures that the data in the data base is valid and correct. Keys play an important role in maintaining data integrity.
The various type of keys that have been identified are given as :
(1) Candidate key
(2) Primary key
(3) Alternate key
(4) Composite key
(5) Unique key
(6) Foreign key
E.g :
In given table, (employee table) there are three attributes (ID, Name, Company).
Here ID is a primary key because there is no repeating value or null value.
Here primary key is used to uniquely identify each row.

Employee :

ID	Name	Company
001	SAIF NAQVI	OSWAL
002	YASMEEN	SONI
003	SHILPI	BPL
004	GOURAV	HCL
005	SHILPI	NOKIA
006	PANKAJ	SONI

(b) (i) SELECT DCODE, DESCRIPTION FROM DRESS ORDER BY DCODE ASC;

      OR

SELECT DCODE, DESCRIPTION FROM DRESS ORDER BY DCODE;

(Note—If ASC or DESC is not given, SQL assumed as a ASC (*i.e.* ascending) by default)

(ii) SELECT * FROM DRESS WHERE LAUNCHDATE BETWEEN 05–DEC–07 AND 20–JUN–08;

(iii) SELECT AVG (PRICE) FROM DRESS WHERE MCODE = "MOO3";

(iv) SELECT MAX (PRICE), MIN (PRICE), MCODE FROM DRESS, MATERIAL WHERE DRESS. MCODE = MATERIAL. MCODE;

(v) SUM (PRICE)

2700

(vi) **ERROR**

Two tables will be join only on common attributes.

Here attribute DCODE of DRESS and attribute MCODE of MATERIAL are not common on both tables.

(vii) **ERROR**

MAX() function always apply only on Numeric data value. Here "MCODE" is not numeric.

(viii) **COUNT (DISTINCT PRICE)**

6

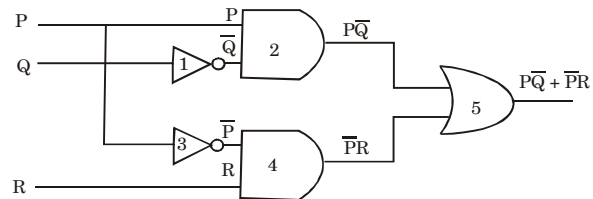**6. (a) Absorption law :**

$x + xy = x$

Truth table for Absorption Law

x	y	xy	$x+xy$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Column No. 1 and 4 have identical value.

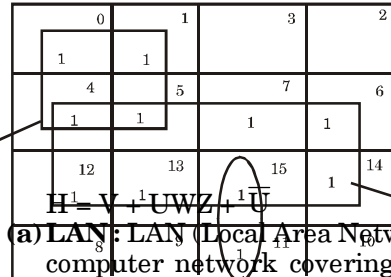Hence, $x + xy = x$

(b)

1—NOT gate

2—AND gate

3—NOT gate

4—AND gate

5—OR gate

Ans. $P\overline{Q} + \overline{P}R$

(c) $G = F (U, V, W) = M_2. M_3. M_6$

$= \pi (2, 3, 6)$

$= (U + \overline{V} + W) (U + \overline{V} + \overline{W})$

$(\overline{U} + \overline{V} + W)$

(d) $H (U, V, W, Z) = \Sigma (0, 1, 4, 5, 6, 7, 11, 12, 13, 14, 15)$

$H = V + UWZ + \overline{U}$

7. (a) LAN : LAN (Local Area Network) is a computer network covering a small geographic area, like a home, office, or group of building.

Network in an organization can be a LAN.

Typically owned, controlled, and managed by a single person or organization.

LANs have a high data transfer rate.

Have small geographical range and do not need any leased telecommunication lines.

One LAN can be connected to other LANS over distance via telephone lines and radio waves.

WAN

WAN (Wide Area Network) is a computer network that covers a broad area (*i.e.*, any network whose communications links cross metropolitan, regional, or national boundaries.

Internet is the best example of WAN. WANs (like the Internet) are not owned by any one organization but rather exist under collective or distributed ownership and management.

WANs have a lower data transfer rate as compared to LANs.

Have a large geographical range generally spreading across boundaries and need leased telecommunication lines.

Computers connected to a wide area network are often connected through public networks, such as the telephone system. They can also be connected through leased lines or satellites.

(b) (i) In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET; a small network of connected computers. The idea was that each host computer (not necessary from the same manufacture) would be attached to a specialized computer, called an interface message processor (IMP). The IMPs, in turn, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles.

(UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via IMPs to form a network.

TCP/IP was Added :
Over the next decade, ARPA net spawned other networks, and in 1983 with more than 300 computers connected, its protocols were changed to TCP/IP. In that same year, the unclassified military Milnte network was split off from ARPAnet.

It Became the Internet :
As TCP/IP and gateway technologies matured, more disparate networks were connected, and the ARPAnet became known as "the Internet" and "the Net". Starting in 1987, the National Science Foundation began developing a high-speed backbone between its supercomputer centers. Intermediate networks of regional ARPAnet sites were formed to hook into the backbone, and commercial as well a non-profit network service providers were formed to handle the operations. Over time, other federal agencies and organizations formed backbones that linked in.

(ii) **HTTP**

Hyper Text Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. Its use for retrieving inter-linked resources lead to the establishment of the World Wide Web.

HTTP is a "stateless" request/response system. The connection is maintained between client and server only for the immediate request and the connection is closed. After the HTTP client establishes a TCP connection with the server and sends it a request command, the server sends back its response and close the connection.

HTTP development was coordinated by the World Wide Web Consortium and the Internet Engineering Task Force (IETF). HTTP is a request/response standard of a client and a server. A client is the end-user, the server is the web site. The client making a HTTP request—using a web browser, spider, or other end-user tool—is referred to as the *user agent*. The responding server—which stores or creates *resources* such as HTML files and images—is called the *origin server*. In between the user agent and origin server may be several intermediaries, such as proxies, gateways, and tunnels. HTTP is not constrained to using TCP/IP and its supporting layers, although this is its most popular

application on the Internet. Indeed HTTP can be "implemented on top of any other protocol on the Internet, or on other networks." HTTP only presumes a reliable transport; any protocol on the provides such guarantees can be used. Typically, an HTTP client initiates a request. It establishes a Transmission Control Protocol (TCP) connection to a particular port on a host (port 80 by default). An HTTP server listening on that port waits for the client to send a request message. Upon receiving the request, the server sends back a status line, such as "HTTP/ 1.1200 OK", and a message of its own, the body of which is perhaps the requested resource, an error message, or some other information.

Resources to be accessed by HTTP are identified using Uniform Resource Identifiers (URIs) (or, more specifically, Uniform Resource Locators (URLs)) using the http: or https URI schemes.

(c) Protocol

Protocol is a set of "RULES" and "REGULATIONS" for sending and receiving information on the NETWORK, by using the standard protocols, TCP/IP.

A protocol is set of rules or agreed upon guidelines for communication. When communicating, it is important to agree on how to do so. If one party speaks Indian and one German, the communications will most likely fail. If they both agree on single language, communication will work. On the Internet, the set of communications protocols used is called TCP/IP. TCP/ IP is actually a collection of various protocols that each have their own special function or purpose. These protocols have been established by international standards bodies and are used in almost all platforms and around the globe to ensure that all devices on the Internet can communicate successfully.

File Transfer Protocol (FTP) is used to copy a file from/to a remotely located server.

(d) (1) Circuit switching:
- Path to be decided before data transmission starts.
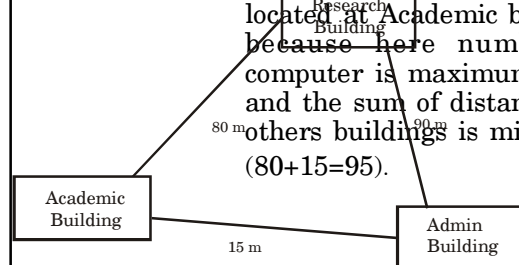- So less overhead for data packets for routing it.

(2) Packet switching:
- At the time of connection no need to worry about routing.
- Ability of route data units over any route, so more reliable.

(e) (E–1) Since. here university campus is LAN so we will use Unshielded Twisted Pair (UTP) cable.

(E–2) Server will be located at which building where number of computers will be maximum and the sum of distances for other buildings (within University) will be minimum. Hence, it (server) must be located at Academic building because here number of computer is maximum (150) and the sum of distances for others buildings is minimum (80+15=95).

(E–3) (iii) SWITCH

(E–4) Leased line.