1.Develop a simple library management system with class name **LIBRARY** and members as shown below. At the beginning program should accept values for data members for atleast 10 books in the library.

Class name: Library

Struct {
Data member:
Acc No. , Title of book, Author, Status, USN, Name }

Member functions: read info( ) write info( ) search ( ) issue ( ) / return( )

Status: Available------------------>1
Not Available----------->0

After reading the information about books, the program should allow the librarian to issue or receive the books and to display the status of the books

Program:

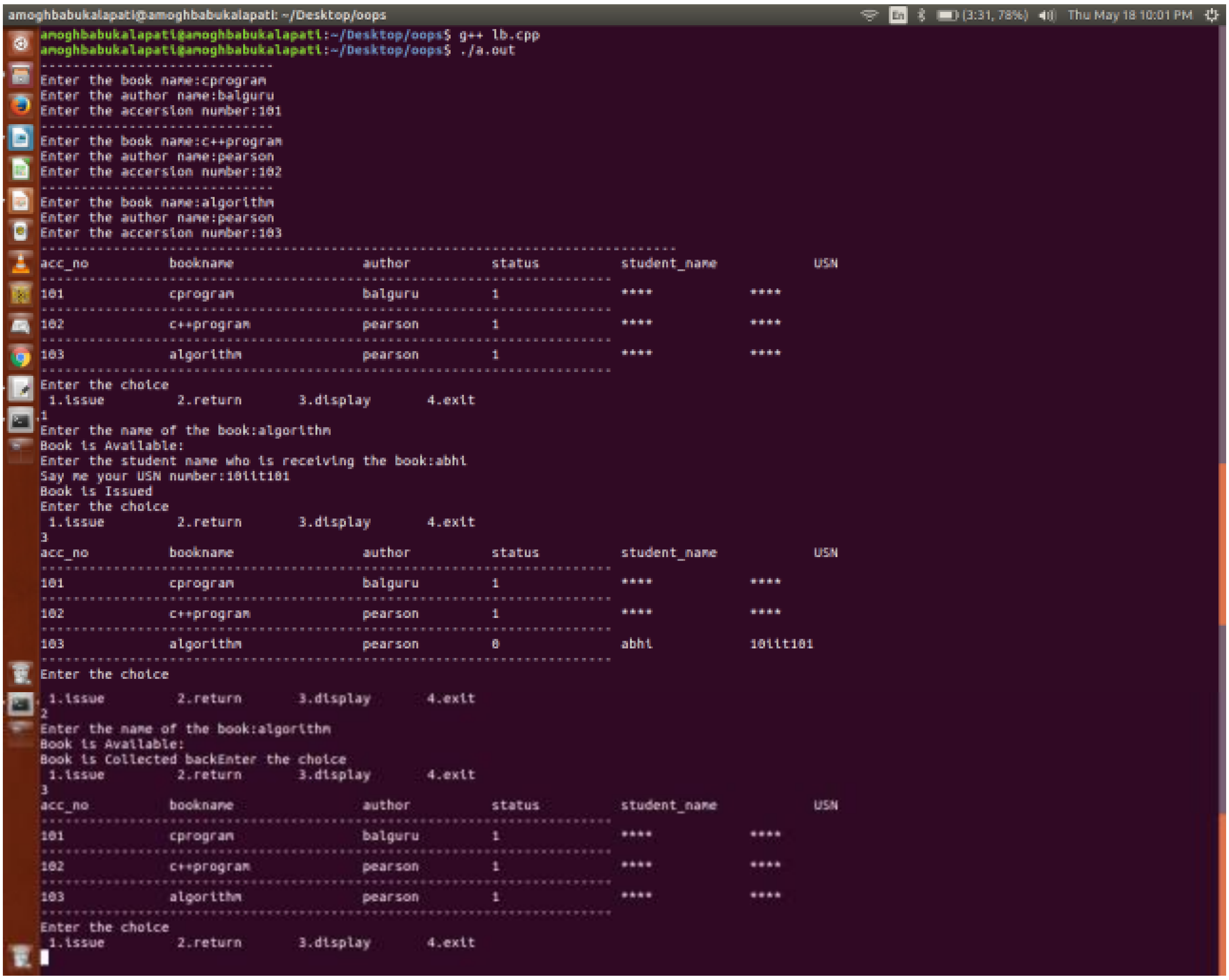
```
#include<string.h>
#include<stdlib.h>
#include<iostream.h>
class library
{
        struct details
        {
                int status,acc_no;
                char std_name[20] ,usn[20],  author[20],bookname[20];
        }d[20];
        int nb;
        public:
        library (int x)
        {
                nb=x;
                for(int i=0;i<nb;i++)
                {
                        d[i].status=1;
                        strcpy(d[i].usn,"****");
                        strcpy(d[i].std_name,"****");
                }
        }
```

```
void read()
{
        for(int i=0;i<nb;i++)
        {
            cout<<"----------------------------"<<endl;

            cout<<"Enter the book name:";
            cin>>d[i].bookname;
            cout<<"Enter the author name:";
            cin>>d[i].author;
            cout<<"Enter the accersion number:";
            cin>>d[i].acc_no;
        }
}
void issue(int i)
{
    char std_name1[20],usn1[20];
    cout<<"Enter the student name who is receiving the book:";
    cin>>std_name1;
    cout<<"Say me your USN number:";
    cin>>usn1;
    d[i].status=0;
    strcpy(d[i].std_name,std_name1);
    strcpy(d[i].usn,usn1);
    cout<<"Book is Issued"<<endl;
}
void return_book(int i)
{
        d[i].status=1;
        strcpy(d[i].std_name,"****");
        strcpy(d[i].usn,"****");
        cout<<"Book is Collected back";
}
void  display()
{
    int i;
```

```
                for(i=0;i<nb;i++)
                {       cout<<"----------------------------------------------------------
------------"<<endl;
cout<<d[i].acc_no<<"\t"<<d[i].bookname<<"\t\t"<<d[i].author<<"\t"<<d[i
].status<<"\t"<<d[i].std_name<<"\t\t"<<d[i].usn<<endl;
                }
        }
        int search()
        {
                char bname[20];
                cout<<"Enter the name of the book:";
                cin>>bname;
                for(int i=0;i<nb;i++)
                {
                        if(strcmp(bname,d[i].bookname)==0)
                        {
                                cout<<"Book is Available:"<<endl;
                                return (i);
                        }
                }
                return -1;
        }
};
void main()
{
        library l(3);
        int ch,x,n;
       char bname[20];
        l.read();
cout<<"------------------------------------------------------------------------------
"<<endl;
cout<<"acc_no\t"<<"bookname\t"<<"author\t"<<"status\t"<<"student_na
me \t"<<"USN"<<endl;
        l.display();
        while(1)
        {
```

```
        cout<<"Enter the choice " << endl << " 1 . i ssue\t
2.return\t3.display\t4.exit\n";
                cin>>ch;
                switch(ch)
                {
                        case 1: x=l.search();
                                l.issue(x);
                                break;
                        case 2: x=l.search();
                                l.return_book(x);
                                break;
                        case                                            3:
cout<<"acc_no\t"<<"bookname\t"<<"author\t"<<"status\t"<<"student_na
me \t"<<"USN"<<endl;
                                l.display();
                                break;
                        case 4:
                                exit(1);
                 }
        }
        getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ g++ lb.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
-----------------------------
Enter the book name:cprogram
Enter the author name:balguru
Enter the accersion number:101
-----------------------------
Enter the book name:c++program
Enter the author name:pearson
Enter the accersion number:102
-----------------------------
Enter the book name:algorithm
Enter the author name:pearson
Enter the accersion number:103
-------------------------------------------------------------------------------
acc_no          bookname                author          status          student_name            USN
-----------------------------------------------------------------------
101             cprogram                balguru         1               ****            ****
-----------------------------------------------------------------------
102             c++program              pearson         1               ****            ****
-----------------------------------------------------------------------
103             algorithm               pearson         1               ****            ****
-----------------------------------------------------------------------
Enter the choice
 1.issue         2.return       3.display       4.exit
1
Enter the name of the book:algorithm
Book is Available:
Enter the student name who is receiving the book:abhi
Say me your USN number:10iit101
Book is Issued
Enter the choice
 1.issue         2.return       3.display       4.exit
3
acc_no          bookname                author          status          student_name            USN
-----------------------------------------------------------------------
101             cprogram                balguru         1               ****            ****
-----------------------------------------------------------------------
102             c++program              pearson         1               ****            ****
-----------------------------------------------------------------------
103             algorithm               pearson         0               abhi            10iit101
-----------------------------------------------------------------------
Enter the choice

 1.issue         2.return       3.display       4.exit
2
Enter the name of the book:algorithm
Book is Available:
Book is Collected backEnter the choice
 1.issue         2.return       3.display       4.exit
3
acc_no          bookname                author          status          student_name            USN
-----------------------------------------------------------------------
101             cprogram                balguru         1               ****            ****
-----------------------------------------------------------------------
102             c++program              pearson         1               ****            ****
-----------------------------------------------------------------------
103             algorithm               pearson         1               ****            ****
-----------------------------------------------------------------------
Enter the choice
 1.issue         2.return       3.display       4.exit
```

2.Develop a program to generate cinema bill with a class name cinema and data members as date , time, number of adults, number of children and bill amount, theatre name and cinema name. Overload the function by name **C_BILL** to generate the bill for the following categories

a. Only adult
b. Only children
c. Both adult and children

Call system data and time.

Program:

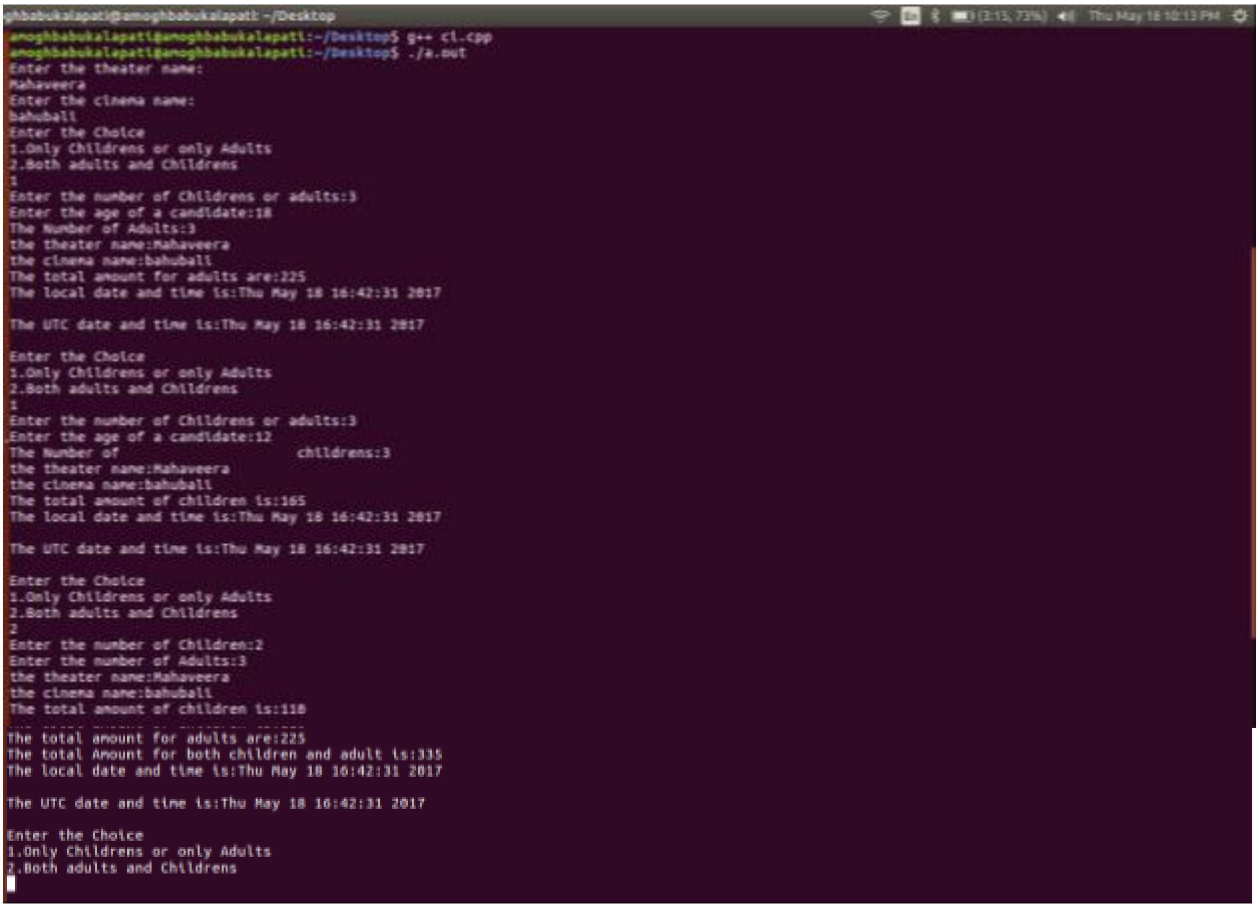
```
#include<iostream.h>
#include<conio.h>
```

```
#include <time.h>
#include<stdlib.h>
#define amt_ch 55
#define amt_adt 75
class cinema
{
        int adt,ch,y;
        int i,h,n,z,x;
        char t[10],c[10],dt;
        //char*dt;
        public:
        void read()
        {
                cout<<"Enter the theater name:"<<endl;
                cin>>t;
                cout<<"Enter the cinema name:"<<endl;
                cin>>c;
        }
        void c_bill(int ch)
        {
                cout<<"Enter the age of a candidate:";
                cin>>x;
                if(x<18)
                {
                        z=ch*amt_ch;
                        cout<<"The Number of 
childrens:"<<ch<<endl;
                        display1();
                }
                else
                {
                        adt=ch;
                        n=adt*amt_adt;
                        cout<<"The Number of Adults:"<<adt<<endl;
                        display2();
```

```
		}
	}
	void c_bill(int ch,int adt)
	{
		z=ch*amt_ch;
		n=adt*amt_adt;
		i=n+z;
		display();
	}
	void display()
	{
		cout<<"the theater name:"<<t<<endl;
		cout<<"the cinema name:"<<c<<endl;
		cout<<"The total amount of children is:"<<z<<endl;
		cout<<"The total amount for adults are:"<<n<<endl;
		cout<<"The total Amount for both children and adult
is:"<<i<<endl;
	}
	void display1()
	{
		cout<<"the theater name:"<<t<<endl;
		cout<<"the cinema name:"<<c<<endl;
		cout<<"The total amount of children is:"<<z<<endl;
	}
	void display2()
	{
		cout<<"the theater name:"<<t<<endl;
		cout<<"the cinema name:"<<c<<endl;
		cout<<"The total amount for adults are:"<<n<<endl;
	}
};
void main()
{
	cinema ci;
	int choice,ch,ad;
```

```
        char *dt=NULL;
        ci.read();
        time_t now = time(0);
        dt=ctime(&now);
        tm *gmtm = gmtime(&now);
        dt = asctime(gmtm);
        cout<<"Enter the Choice"<<endl;
        cout<<"1.Only Childrens or only Adults\n2.Both adults and 
Childrens"<<endl;
        cin>>choice;
        switch(choice)
        {
                case 1:cout<<"Enter the number of Childrens or adults:";
                        cin>>ch;
                        ci.c_bill(ch);
                        cout<<"The local date and time is:"<<dt<<endl;
                        cout<<"The UTC date and time is:"<<dt<<endl;
                        break;

                case 2:cout<<"Enter the number of Children:";
                        cin>>ch;
                        cout<<"Enter the number of Adults:";
                        cin>>ad;
                        ci.c_bill(ch,ad);
                        cout<<"The local date and time is:"<<dt<<endl;
                        cout<<"The UTC date and time is:"<<dt<<endl;
                        break;
                default:cout<<"Enter the valid Choice:";
        }
        getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop$ g++ c1.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop$ ./a.out
Enter the theater name:
Mahaveera
Enter the cinema name:
bahuball
Enter the Choice
1.Only Childrens or only Adults
2.Both adults and Childrens
1
Enter the number of Childrens or adults:3
Enter the age of a candidate:18
The Number of Adults:3
the theater name:Mahaveera
the cinema name:bahuball
The total amount for adults are:225
The local date and time is:Thu May 18 16:42:31 2017

The UTC date and time is:Thu May 18 16:42:31 2017

Enter the Choice
1.Only Childrens or only Adults
2.Both adults and Childrens
1
Enter the number of Childrens or adults:3
Enter the age of a candidate:12
The Number of                    childrens:3
the theater name:Mahaveera
the cinema name:bahuball
The total amount of children is:165
The local date and time is:Thu May 18 16:42:31 2017

The UTC date and time is:Thu May 18 16:42:31 2017

Enter the Choice
1.Only Childrens or only Adults
2.Both adults and Childrens
2
Enter the number of Children:2
Enter the number of Adults:3
the theater name:Mahaveera
the cinema name:bahuball
The total amount of children is:110
The total amount for adults are:225
The total Amount for both children and adult is:335
The local date and time is:Thu May 18 16:42:31 2017

The UTC date and time is:Thu May 18 16:42:31 2017

Enter the Choice
1.Only Childrens or only Adults
2.Both adults and Childrens
```

3.Create a two classes **DM** and **DF** which stores the value of distance. DM stores distance in meters and centimeters, DF in feet and inches. Write a program that can read values for class objects and add one objects of DM with another object DF. Display the results as per user choice.

Program:

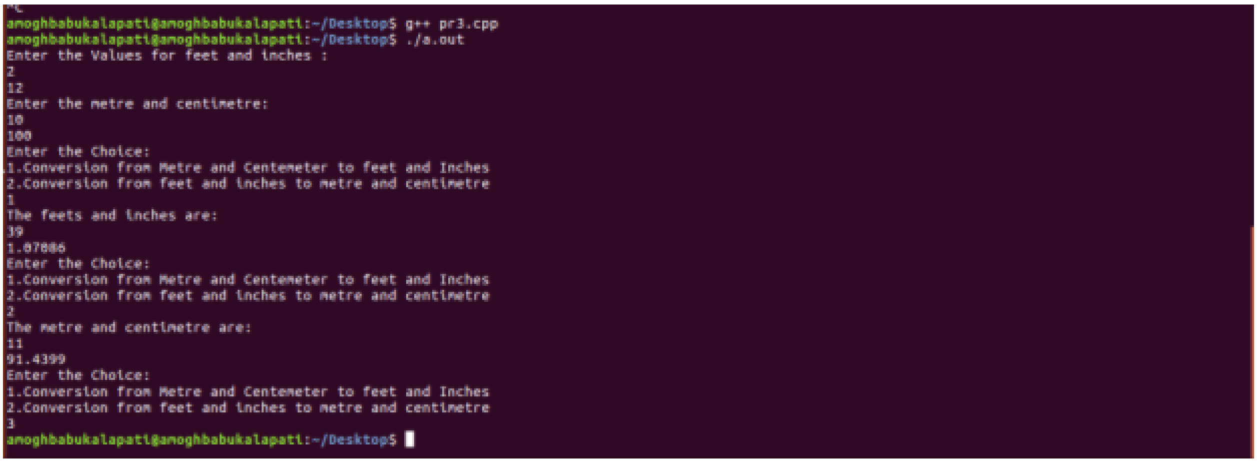
```
#include<math.h>
#include<iostream.h>
#include<stdlib.h>
class DM;
class DF
{
        float inch,feet;
        public:
        int getdata()
        {
                cout<<"Enter the Values for feet and inches :"<<endl;
                cin>>feet>>inch;
```

```
	}
	int display()
	{
		cout<<"The feets and inches are:"<<endl;
		cout<<feet<<endl<<inch<<endl;
	}
	friend DF compute(DF,DM);
	friend DM compute1(DF,DM);
};
class DM
{
	float mt,cm;
	public:
	int getdata()
	{
		cout<<"Enter the metre and centimetre:"<<endl;
		cin>>mt>>cm;
	}
	int display()
	{
		cout<<"The metre and centimetre are:"<<endl;
		cout<<mt<<endl;
		cout<<cm<<endl;
	}
	friend DF compute(DF,DM);
	friend DM compute1(DF,DM);
};
DF compute(DF x,DM y)
{
	int count=0;
	y.cm=y.cm+y.mt*100;
	x.inch=x.inch+x.feet*12+y.cm/2.54;
	while(x.inch>=12)
	{
		x.inch=x.inch-12;
		count++;
```

```
    }
    x.feet=count;
    return(x);
}
DM compute1(DF a,DM b)
{
    int count=0;
    b.cm=b.cm+b.mt*100+a.inch*2.54+a.feet*12*2.54;
    while(b.cm>=100)
    {
        b.cm=b.cm-100;
        count++;
    }
    b.mt=count;
    return(b);
}
void main()
{
    DF p;
    DM s;
    DM q;
    DF r;
    int ch;
    p.getdata();
    q.getdata();
    while(1)
    {
        cout<<"Enter the Choice:"<<endl<<"1.Conversion from Metre and Centemeter to feet and Inches"<<endl<<"2.Conversion from feet and inches to metre and centimetre"<<endl;
        cin>>ch;
        switch(ch)
        {
            case 1:r=compute(p,q);
                r.display();
                break;
```

```
                    case 2:s=compute1(p,q);
                          s.display();
                          break;
                    default:exit(0);
               }
          }
          getch();
}
```

Run:



```
^C
amoghbabukalapati@amoghbabukalapati:~/Desktop$ g++ pr3.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop$ ./a.out
Enter the Values for feet and inches :
2
12
Enter the metre and centimetre:
10
100
Enter the Choice:
1.Conversion from Metre and Centemeter to feet and Inches
2.Conversion from feet and inches to metre and centimetre
1
The feets and inches are:
39
1.07086
Enter the Choice:
1.Conversion from Metre and Centemeter to feet and Inches
2.Conversion from feet and inches to metre and centimetre
2
The metre and centimetre are:
11
91.4399
Enter the Choice:
1.Conversion from Metre and Centemeter to feet and Inches
2.Conversion from feet and inches to metre and centimetre
3
amoghbabukalapati@amoghbabukalapati:~/Desktop$ 
```

4.Develop a program to monitor the status of a 2 conference hall with respect to its capacity. The maximum seating capacity of each hall in 30. Create a class by name conference_hall with the following data members

a. Name of the conference hall
b. Availability of seats

Update seat availability in each hall as and when the delegates enter and leave the hall.

Write the following member functions

i. enter_hall( ) – overload unary +
ii. exit_hall( ) - overload unary -
iii. Overload << operator to display the number of seats available in conference halls
iv. Overload the binary + operator to display the total no. of available seats in H1 and H2

Program:

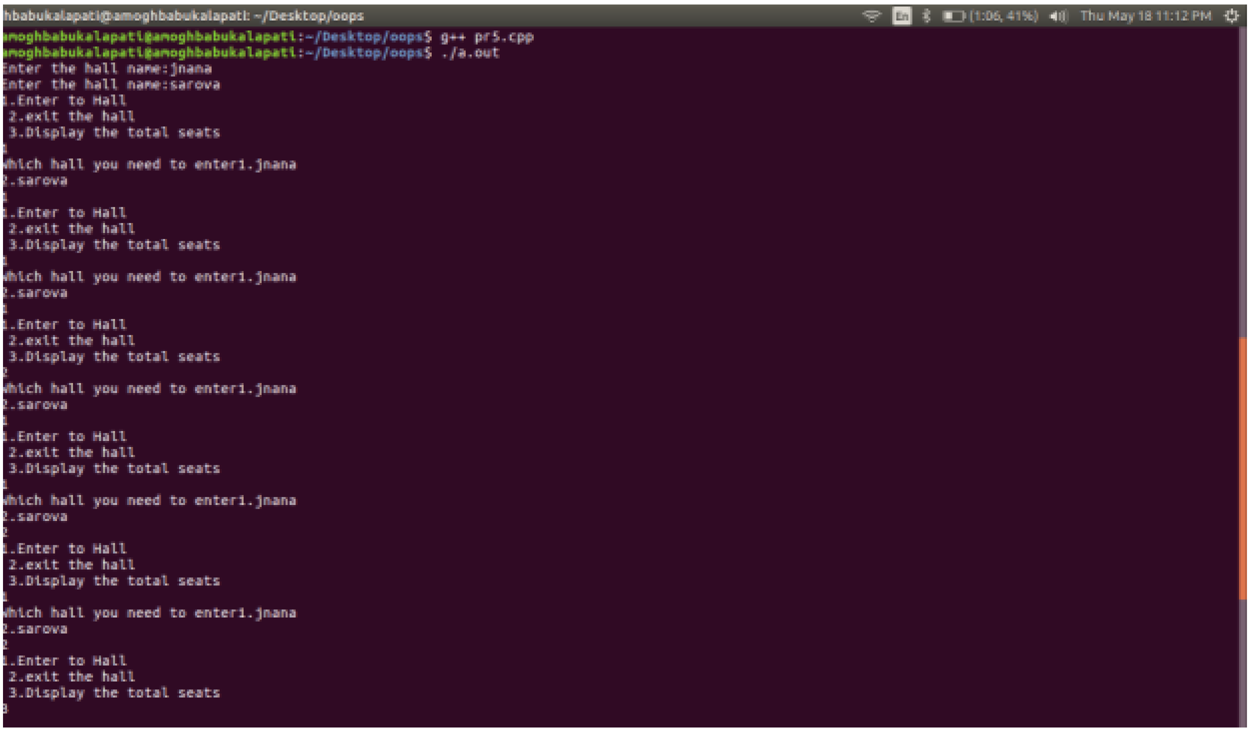
```
#include<iostream.h>
#include<stdlib.h>
```

```
#include<string.h>
class conferencehall
{
	int avail;
	char name[20];
	public:
	conferencehall()
	{
		avail=30;
	}
	void enterhall()
	{
		if((avail>=1)&&(avail<=30))
		avail--;
	}
	void exithall()
	{
		avail++;
		if(avail>30)
		avail--;
	}
	void accept()
	{
		cout<<"Enter the hall name:";
		cin>>name;
	}
	int operator+(conferencehall b)
	{
		int p;
		p=b.avail+avail;
		return p;
	}
	void display()
	{
		cout<<"The Available seat Capacity in "<<name<<"hall
is"<<avail<<endl;
```

```
	}
	void display1()
	{
		cout<<name<<endl;
	}
	friend ostream &operator <<(ostream &,conferencehall &);
	friend istream &operator >>(istream &,conferencehall &);
	friend void write(conferencehall,conferencehall);
};
void write(conferencehall x,conferencehall y)
{
	int s;
	s=x+y;
	x.display();
	y.display();
	cout<<"The Total Seats Available in both the hall is"<<s<<endl;
}
ostream& operator <<(ostream &out,conferencehall &o)
{
	out<<"The number of seats available is";
	out<<o.avail;
	return out;
}
istream& operator >>(istream &in,conferencehall &o)
{
	in>>o.avail;
	return in;
}
void main()
{
	int n,x;
	conferencehall h1,h2;
	h1.accept();
	h2.accept();
	while(1)
	{
```

```
                cout<<"1.Enter Hall\n 2.Exit hall \n 3.Display total seats\n";
                cin>>n;
                switch(n)
                {
                        case 1: cout<<"Which hall you need to enter";
                                cout<<"1.";
                                h1.display1();
                                cout<<"2.";
                                h2.display1();
                                cin>>x;
                                if(x==1)
                                h1.enterhall();
                                else
                                h2.enterhall();
                                break;
                        case 2:      cout<<"Which hall you need to enter";
                                cout<<"1.";
                                h1.display1();
                                cout<<"2.";
                                h2.display1();
                                cin>>x;
                                if(x==1)
                                h1.exithall();
                                else
                                h2.exithall();
                                break;
                        case 3: write(h1,h2);
                                break;
                        default:cout<<"Exting................!!!!!";
                                exit(0);

                }
        }
        getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ g++ pr5.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
Enter the hall name:jnana
Enter the hall name:sarova
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
1
Which hall you need to enter1.jnana
2.sarova
1
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
1
Which hall you need to enter1.jnana
2.sarova
1
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
2
Which hall you need to enter1.jnana
2.sarova
1
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
1
Which hall you need to enter1.jnana
2.sarova
2
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
1
Which hall you need to enter1.jnana
2.sarova
2
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
3
```

5. Write a program to create a class called MATRIX using a 2-dimensional array of integers. Implement the following by overloading operators = =, *, << and >> .

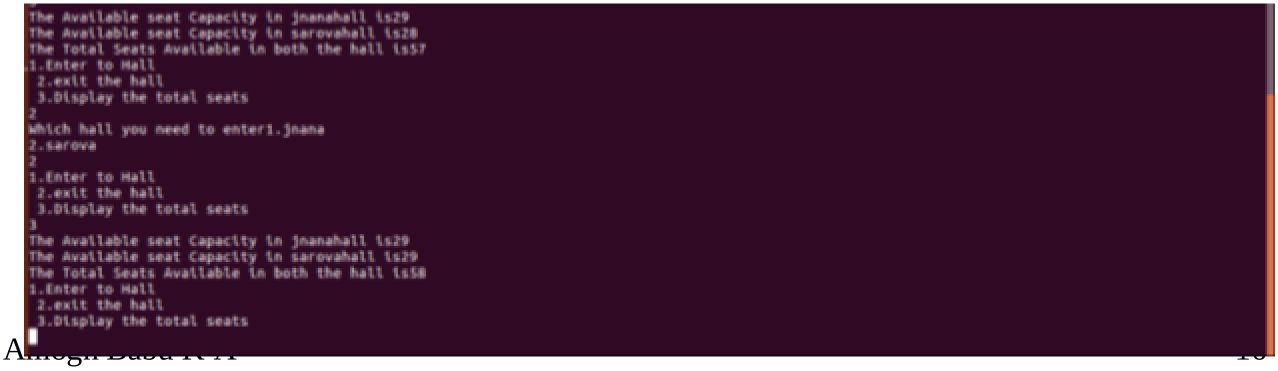
a. M3=M1*M2

b. M4=M2 * c where c is a constant number.

Dynamically allocate memory for matrix.

Program:

```
#include<iostream.h>
#include<stdlib.h>
#include<malloc.h>
class matrix
{
        int r,c,**m;
        public:
```

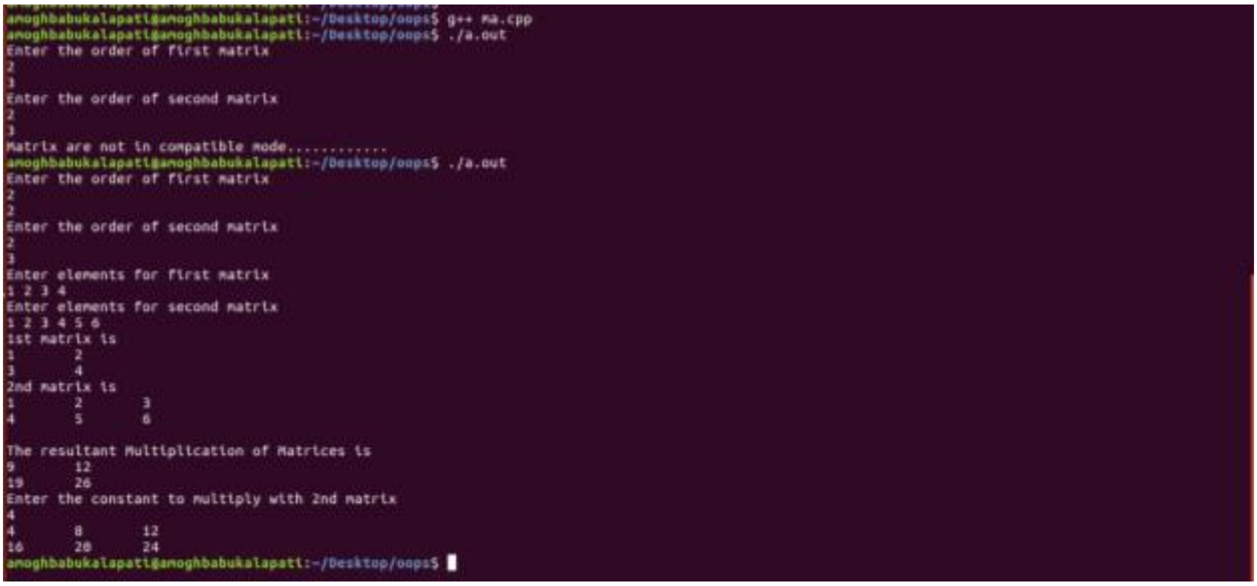


```
The Available seat Capacity in jnanahall is29
The Available seat Capacity in sarovahall is28
The Total Seats Available in both the hall is57
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
2
Which hall you need to enter1.jnana
2.sarova
2
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
3
The Available seat Capacity in jnanahall is29
The Available seat Capacity in sarovahall is29
The Total Seats Available in both the hall is58
1.Enter to Hall
 2.exit the hall
 3.Display the total seats
```

```
        matrix(int x,int y);
        int operator==(matrix);
        matrix operator*(matrix);
        matrix operator*(int);
        friend ostream &operator<<(ostream &out,matrix &);
        friend istream &operator>>(istream &in,matrix &);
};
matrix::matrix(int x,int y)
{
        r=x;
        c=y;
        m=new int *[r];  //It creates an array pointer
        for(int i=0;i<r;i++)
        m[i]=new int [c]; //create space for each row
}
matrix matrix::operator*(int p)
{
        matrix m4(r,c);
        for(int i=0;i<r;i++)
        for(int j=0;j<c;j++)
        m4.m[i][j]=m[i][j]*p;
        return m4;
}
int matrix::operator==(matrix m1)
{
        if(c==m1.r)
        return 1;
        return 0;
}
matrix matrix::operator*(matrix m1)
{
        matrix temp(r,c);
        int i,j,k;
        for(i=0;i<r;i++)
        {
                for(j=0;j<m1.c;j++)
                {
                        temp.m[i][j]=0;
                        for(k=0;k<c;k++)
                        {
                                temp.m[i][j]+=m[i][k]*m1.m[k][j];
                        }
                }
        }
        return temp;
}
ostream &operator<<(ostream &out,matrix &m1)
{
        int i,j;
        for(i=0;i<m1.r;i++)
        {
                for(j=0;j<m1.c;j++)
                {
                        out<<m1.m[i][j]<<"\t";
                }
                out<<endl;
        }
        return(out);
}
istream &operator>>(istream &in,matrix &m1)
{
        int i,j;
        for(i=0;i<m1.r;i++)
        {
                for(j=0;j<m1.c;j++)
                in>>m1.m[i][j];
        }
        return(in);
}
void main()
{
        int m,n,p,q,c;
```

```
	cout<<"Enter the order of first matrix"<<endl;
	cin>>m>>n;
	matrix m1(m,n);
	cout<<"Enter the order of second matrix"<<endl;
	cin>>p>>q;
	matrix m2(p,q);
	if(m1==m2)
	{
		cout<<"Enter elements for first matrix"<<endl;
		cin>>m1;
		cout<<"Enter elements for second matrix"<<endl;
		cin>>m2;
		cout<<"1st matrix is"<<endl;
		cout<<m1;
		cout<<"2nd matrix is"<<endl;
		cout<<m2;
		matrix m3(m,q);
		m3=m1*m2;
		cout<<endl<<"The resultant Multiplication of Matrices
is"<<endl;
		cout<<m3;
	}
	else
	{
		cout<<"Matrix are not in compatible mode............ "<<endl;
		exit(0);
	}
	cout<<"Enter the constant to multiply with 2nd matrix"<<endl;
	cin>>c;
	matrix m4(p,q);
	m4=m2*c;
	cout<<m4;
	getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ g++ ma.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
Enter the order of first matrix
2
3
Enter the order of second matrix
2
3
Matrix are not in compatible mode............
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
Enter the order of first matrix
2
2
Enter the order of second matrix
2
3
Enter elements for first matrix
1 2 3 4
Enter elements for second matrix
1 2 3 4 5 6
1st matrix is
1       2
3       4
2nd matrix is
1       2       3
4       5       6

The resultant Multiplication of Matrices is
9       12
19      26
Enter the constant to multiply with 2nd matrix
4
4       8       12
16      20      24
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ 
```

6.Write a program to perform search operation on a given list of data (**int or float or char**) . If the given list is sorted perform binary search else perform linear search.

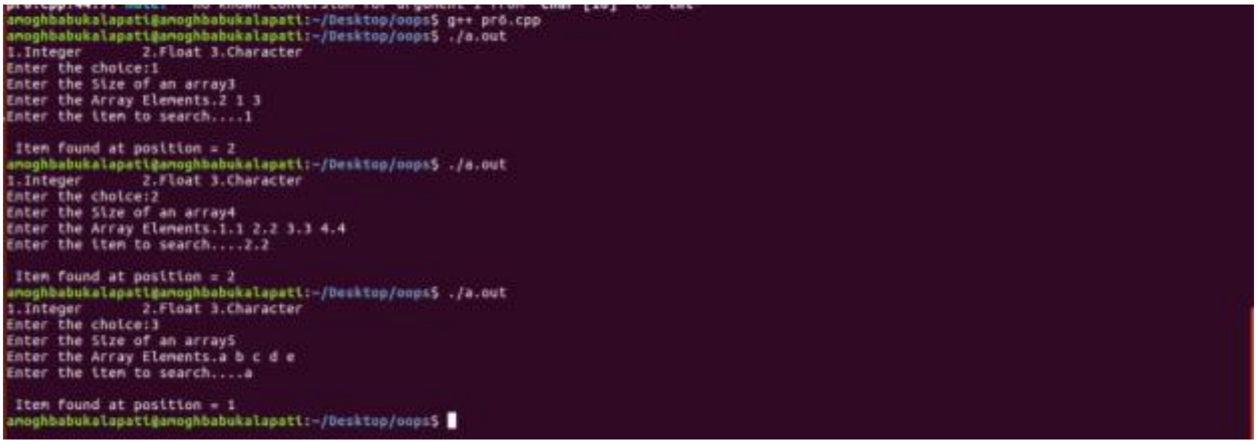
Program:

```
#include<iostream.h>
#include<stdlib.h>
template <class F>
class searching
{
        F a[10];
        public:
        bool is_sorted (F a[20],int n)
        {
                int j;
                for(j=0;j<=n-1;j++)
                {
                        if(a[j]>a[j+1])
                        return (1);
                }
                return (0);
        }
        void binary(F a[20],F n,F t)
        {
                int first,last,middle;
                first=0;
                last=n-1;
                while(first<=last)
                {
                        middle=(first+last)/2;
                        if(a[middle]<t)
                        {
                                first = middle+1;
                        }
                        else if(a[middle]==t)
                        {
                                cout<<t<<" found at location "<<middle+1<<"\n";
                                break;
                        }
                        else
                        {
                                 last = middle - 1;
                        }
                }
                cout<<"Not found! "<<t<<" is not present in the list.";
        }
        void Lsearch(F *a, F item, int n)
        {
                for(int i=0;i<n;i++)
                {
                        if(a[i]== item)
                        {
                                cout<<"\n Item found at position = "<<i+1<<"\n";
                                exit(0);
                        }
                }
                cout<<"\n Item not found in the list\n\n";
        }
};
void main()
```

```
{
        int ch;
        cout<<"1.Integer\t2.Float\t3.Character\nEnter the choice:";
        cin>>ch;
        if(ch==1)
        {
                searching <int > k;
                int a[10],ele,n;
                cout<<"Enter the Size of an array";
                cin>>n;
                cout<<"Enter the Array Elements.";
                for(int i=0;i<n;i++)
                cin>>a[i];
                cout<<"Enter the item to search....";
                cin>>ele;
                if(k.is_sorted(a,n)==0)
                {
                        k.binary(a,n,ele);
                }
                else
                k.Lsearch(a,ele,n);
        }
        else if(ch==2)
        {
                searching <float > k;
                int n;
                float a[10],ele;
                cout<<"Enter the Size of an array";
                cin>>n;
                cout<<"Enter the Array Elements.";
                for(int i=0;i<n;i++)
                cin>>a[i];
                cout<<"Enter the item to search....";
                cin>>ele;
                if(k.is_sorted(a,n)==0)
                {
```

```
                k.binary(a,n,ele);
            }
            else
            k.Lsearch(a,ele,n);
        }
        else if(ch==3)
        {
            searching <char > k;
            int n;
            char a[10],ele;
            cout<<"Enter the Size of an array";
            cin>>n;
            cout<<"Enter the Array Elements.";
            for(int i=0;i<n;i++)
            cin>>a[i];
            cout<<"Enter the item to search....";
            cin>>ele;
            if(k.is_sorted(a,n)==0)
            {
                k.binary(a,n,ele);
            }
            else
            k.Lsearch(a,ele,n);
        }
        else
        cout<<"Invalid";
        getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ g++ pr6.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
1.Integer       2.Float 3.Character
Enter the choice:1
Enter the Size of an array3
Enter the Array Elements.2 1 3
Enter the item to search....1

 Item found at position = 2
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
1.Integer       2.Float 3.Character
Enter the choice:2
Enter the Size of an array4
Enter the Array Elements.1.1 2.2 3.3 4.4
Enter the item to search....2.2

 Item found at position = 2
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
1.Integer       2.Float 3.Character
Enter the choice:3
Enter the Size of an array5
Enter the Array Elements.a b c d e
Enter the item to search....a

 Item found at position = 1
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ █
```

7.Create a linked list of data type (**int or float or char**) & perform following operations.

a).Insert a node at a given position
b).Delete the node with given data
c).Swap the node information of given two nodes.

Program:

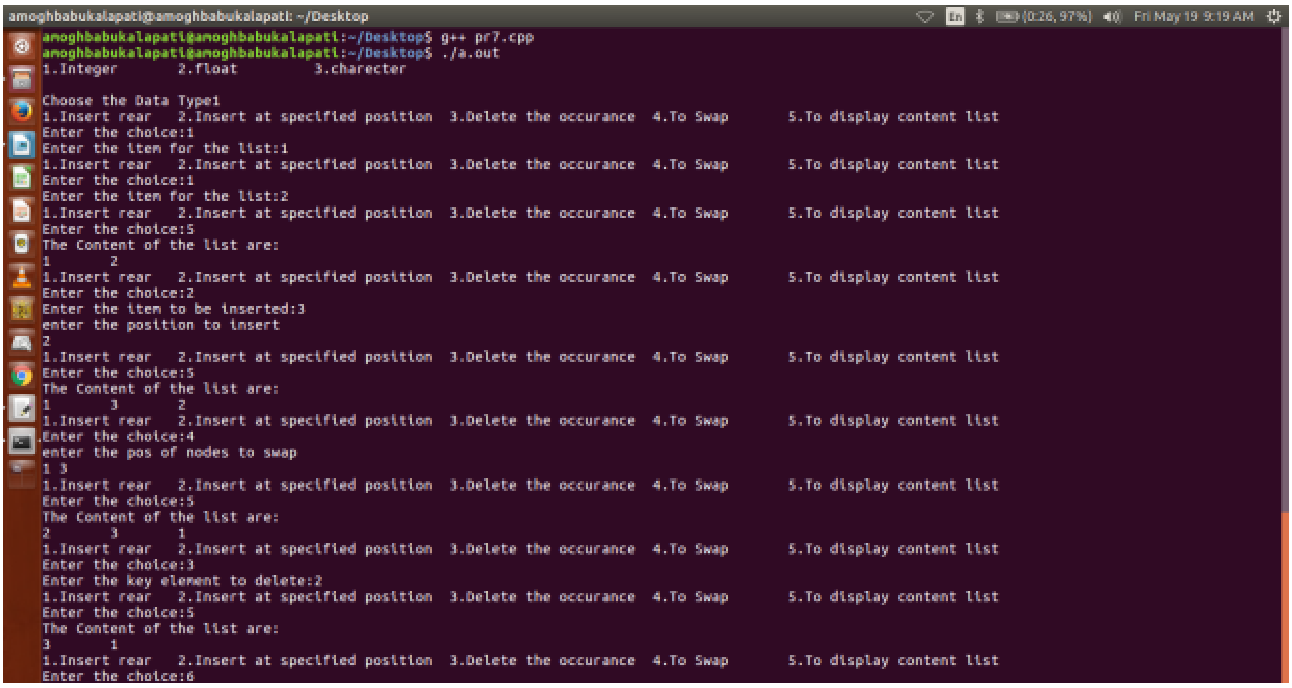
```
#include<iostream.h>
#include<stdlib.h>
template<class t>
class singlelst
{
        struct node
        {
                t info;
                struct node *link;
        };
        typedef struct node *NODE;
        public:
        NODE first=NULL;
        NODE getnode()
        {
                NODE x;
                x=(NODE)malloc(sizeof(struct node));
                if(x==NULL)
                {
                        cout<<"Out of memory...........";
                        exit(0);
                }
                return x;
        }
        NODE insert_rear(t item,NODE first)
        {
                NODE temp;
                NODE cur;
                temp=getnode();
                temp->info=item;
                temp->link=NULL;
                if(first==NULL)
                return temp;
                cur=first;
                while(cur->link!=NULL)
                {
                        cur=cur->link;
                }
                cur->link=temp;
                return first;
        }

        NODE insert_pos(t item,NODE first)
        {
                NODE cur,temp,prev;
                int pos,count=0;
                temp=getnode();
                temp->info=item;
                temp->link=NULL;
                cur=first;
                prev=NULL;
                cout<<"enter the position to insert\n";
                cin>>pos;
                if(pos==1)
                {
                        temp->link=first;
                        first=temp;
                        return first;
                }
                while(cur!=NULL)
                {
                        count++;
                        if(pos==count)
                        {
                                temp->link=cur;
                                prev->link=temp;
                                return first;
                        }
                        prev=cur;
                        cur=cur->link;
                }
                cout<<"position not found\n";
                return first;
        }
        NODE remove(NODE head_ref,t key)
        {
                NODE temp =head_ref, prev;
                while(temp!=NULL&&temp->info== key)
                {
                head_ref=temp->link;
                free(temp);
                temp=head_ref;
                }
                while (temp!=NULL)
                {
                while (temp!=NULL && temp->info!=key)
                {
                                prev=temp;
                                temp=temp->link;
                }
                if(temp==NULL)
                        {
```

```
                        cout<<"Not Possible to delete...!!"<<endl;
                        return head_ref;
                    }
            prev->link=temp->link;
            free(temp);
            temp = prev->link;
            }
    }
    NODE swap(NODE head)
    {
            NODE cur1,cur2;
            int p1,p2,count=0;
            cout<<"enter the pos of nodes to swap\n";
            cin>>p1>>p2;
            cur1=head;
            cur2=head;
            while(cur1!=NULL)
            {
                    count++;
                    if(count==p1)
                    break;
                    cur1=cur1->link;
            }
            count=0;
            while(cur2!=NULL)
            {
                    count++;
                    if(count==p2)
                    break;
                    cur2=cur2->link;
            }
            if(cur1 && cur2 != NULL)
            {
                    t temp;
                    temp=cur1->info;
                    cur1->info=cur2->info;
```

```
                        cur2->info=temp;
                        return head;
                }
                cout<<"pos not found not possible to swap  \n";
                return head;
        }
        void display(NODE first)
        {
                NODE cur;
                if(first==NULL)
                {
                        cout<<"List is Empty.............";
                        return;
                }
                cout<<"The Content of the list are:"<<endl;
                cur=first;
                while(cur!=NULL)
                {
                        cout<<cur->info<<"\t";
                        cur=cur->link;
                }
                cout<<endl;
        }
};
int main()
{
        int ch,choice,pos;
        cout<<"1.Integer\t2.float\t\t3.charecter\n"<<endl;
        cout<<"Choose the Data Type";
        cin>>ch;
        if(ch==1)
        {
                singlelst<int>k;
                int item;
                while(1)
                {
```

```
				cout<<"1.Insert rear\t2.Insert at specified 
position\t3.Delete the occurance\t4.To Swap\t5.To display content list\n";
				cout<<"Enter the choice:";
				cin>>choice;
				switch(choice)
				{
					case 1: cout<<"Enter the item for the list:";
						cin>>item;
						k.first=k.insert_rear(item,k.first);
						break;
					case 2: cout<<"Enter the item to be inserted:";
						cin>>item;
						k.first=k.insert_pos(item,k.first);
						break;
					case 3: cout<<"Enter the key element to delete:";
						cin>>item;
						k.first=k.remove(k.first,item);
						break;
					case 4: k.first=k.swap(k.first);
						break;
					case 5: k.display(k.first);
						break;
					default:
						exit(0);
				}
			}
		}
		else if(ch==2)
		{
			singlelst<float>k;
			float item;
			while(1)
			{
				cout<<"1.Insert rear\t2.Insert at specified 
position\t3.Delete the occurance\t4.To Swap\t5.To display content list\n";
				cout<<"Enter the choice:";
```

```
                cin>>choice;
                switch(choice)
                {
                        case 1: cout<<"Enter the item for the list:";
                                cin>>item;
                                k.first=k.insert_rear(item,k.first);
                                break;
                        case 2: cout<<"Enter the item to be inserted:";
                                cin>>item;
                                k.first=k.insert_pos(item,k.first);
                                break;
                        case 3: cout<<"Enter the key element to delete:";
                                cin>>item;
                                k.first=k.remove(k.first,item);
                                break;
                        case 4: k.first=k.swap(k.first);
                                break;
                        case 5: k.display(k.first);
                                break;
                        default:
                                exit(0);
                }
        }
    }
    else if(ch==3)
    {
        singlelst<char>k;
        char item;
        while(1)
        {
                cout<<"1.Insert rear\t2.Insert at specified
position\t3.Delete the occurance\t4.To Swap\t5.To display content list\n";
                cout<<"Enter the choice:";
                cin>>choice;
                switch(choice)
                {
                        case 1: cout<<"Enter the item for the list:";
                                cin>>item;
                                k.first=k.insert_rear(item,k.first);
                                break;
                        case 2: cout<<"Enter the item to be inserted:";
                                cin>>item;
                                k.first=k.insert_pos(item,k.first);
                                break;
                        case 3: cout<<"Enter the key element to delete:";
                                cin>>item;
                                k.first=k.remove(k.first,item);
                                break;
                        case 4: k.first=k.swap(k.first);
                                break;
                        case 5: k.display(k.first);
                                break;
                        default:
                                exit(0);
                }
        }
    }
    else
    cout<<"not possible......";
    return (0);
}
Run:
```



```
amoghbabukalapati@amoghbabukalapati:~/Desktop$ g++ pr7.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop$ ./a.out
1.Integer       2.float         3.charecter

Choose the Data Type1
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:1
Enter the item for the list:1
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:1
Enter the item for the list:2
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:5
The Content of the list are:
1       2
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:2
Enter the item to be inserted:3
enter the position to insert
2
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:5
The Content of the list are:
1       3       2
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:4
enter the pos of nodes to swap
1 3
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:5
The Content of the list are:
2       3       1
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:3
Enter the key element to delete:2
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:5
The Content of the list are:
3       1
1.Insert rear   2.Insert at specified position  3.Delete the occurance  4.To Swap       5.To display content list
Enter the choice:6
```

8.Create a class called **VECTOR** with array of characters as data member. Create two vector objects and perform the insertion and deletion operations as follows.

Insert elements into first vector object until it is full. Once it is full insert the data into second vector object. Delete operation must begin from first vector.

Program:

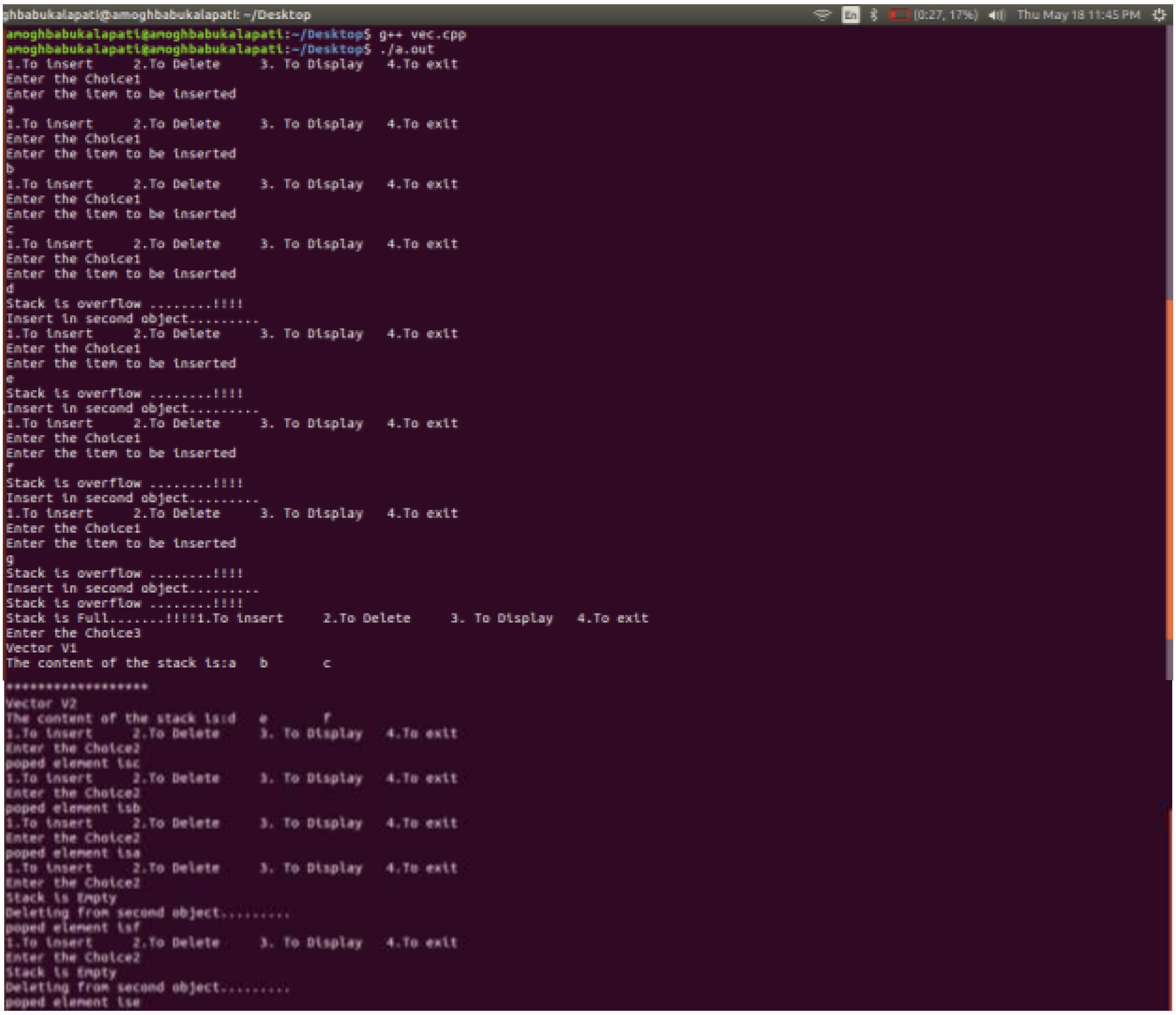
```
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
class vector
{
        char s[10];
        int top,item,size;
        public:
        vector()
        {

                top=-1;
                size=5;
        }
```

```
	int insert(int item1)
	{
		item=item1;
		if(top==size-1)
		{
			cout<<"Stack is overflow ........!!!!"<<endl;
			return 0;
		}
		top=top+1;
		s[top]=item;
		return 1;
	}
	int deletea()
	{
		char num;
		if (top==-1)
		{
			cout<<"Stack is Empty\n";
			return (1);
		}
		else
		{
			num = s[top];
			cout<<"poped element is"<<num<<endl;
			top=top-1;
		}
	}
	void display()
	{
		int i;
		if(top==-1)
		{
			cout<<"Stack is Empty.!!!!!"<<endl;
			return;
		}
		cout<<"The content of the stack is:";
```

```
                for(i=0;i<=top;i++)
                cout<<s[i]<<"\t";
        }
};
void main()
{
        vector v1,v2;
        int choice,s;
        char item;
        while(1)
        {
                cout<<"1.To insert \t2.To Delete \t3. To Display\t4.To
                                exit"<<endl;
                cout<<"Enter the Choice";
                cin>>choice;
                switch(choice)
                {
                        case 1: cout<<"Enter the item to be inserted"<<endl;
                                   cin>>item;
                                   s=v1.insert(item);
                                   if(s==0)
                                   {
                                        cout<<"Insert in second object........."<<endl;
                                        s=v2.insert(item);
                                        if(s==0)
                                        cout<<"Stack is Full.......!!!!"
                                    }
                                    break;
                        case 2: s=v1.deletea();
                                   if(s==1)
                                   {
                                        cout<<"Deleting from second
                                                        object........."<<endl;
                                        s=v2.deletea();
                                        if(s==0)
                                        cout<<"Stack is Empty.......!!!!"
```

```
                  }
                  break;
            case 3: cout<<"Vector V1"<<endl;
                    v1.display();
                    cout<<endl<<"******************"<<endl;
                    cout<<"Vector V2"<<endl;
                    v2.display();
                    cout<<endl;
                    break;
            default:cout<<"Enter the valid choice.............."<<endl;
                    exit(0);
        }
    }
    getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop$ g++ vec.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop$ ./a.out
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
a
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
b
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
c
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
d
Stack is overflow .........!!!!
Insert in second object.........
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
e
Stack is overflow .........!!!!
Insert in second object.........
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
f
Stack is overflow .........!!!!
Insert in second object.........
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice1
Enter the item to be inserted
g
Stack is overflow .........!!!!
Insert in second object.........
Stack is overflow .........!!!!
Stack is Full.......!!!!1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice3
Vector V1
The content of the stack is:a   b       c
******************
Vector V2
The content of the stack is:d   e       f
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice2
poped element isc
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice2
poped element isb
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice2
poped element isa
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice2
Stack is Empty
Deleting from second object.........
poped element isf
1.To insert     2.To Delete     3. To Display   4.To exit
Enter the Choice2
Stack is Empty
Deleting from second object.........
poped element ise
```

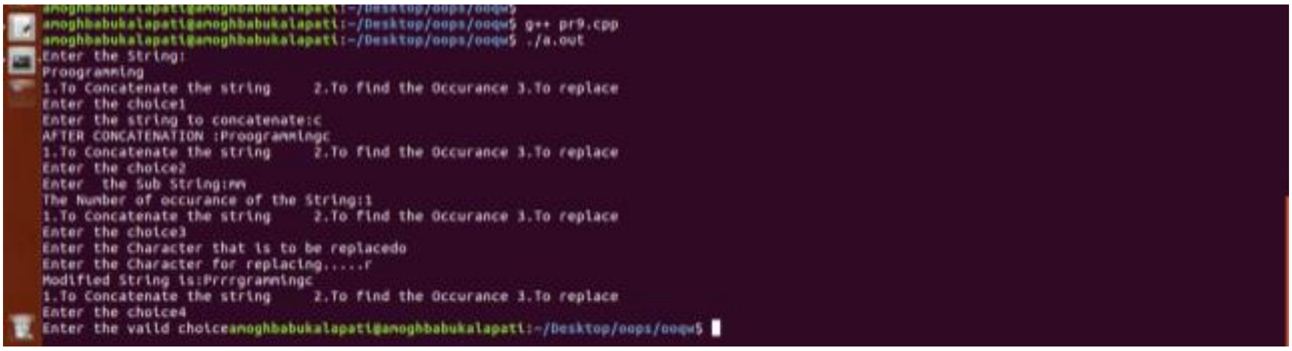
9. Create a class **STRING** and implement the following operations

- **a.** Concatenate 2 string objects using Copy Constructor.
- **b.** Count the number of occurrence of a given sub string in a string.
- **c.** Replace a given character in a string by other character

Program:

```
#include<iostream.h>
#include<string.h>
#include<stdlib.h>
class String
{
        char str[100];
        public:
```

```
String()
{
        int length;
        cout<<"Enter the String:"<<endl;
        cin>>str;
        length=strlen(str);
}
String (char strc[20])
{
        strcpy(str,strc);
}
String(String &s,String &s1)
{
        strcat(s.str,s1.str);
        cout<<"AFTER CONCATENATION :";
        cout<<s.str<<endl;
}
void occuarance()
{
        int count=0,c=0,k,size,length;
        char sub[10];
        cout<<"Enter  the Sub String:";
        cin>>sub;
        length=strlen(str);
        size=strlen(sub);
        for(int i=0;i<=length-size;i++)
        {
                k=0;
                for(int j=0;j<size;j++)
                {
                        if(str[i+j]!=sub[j])
                        {
                                k=1;
                                break;
                        }
                }
                if(k==0)
                count++;
            }
            if(count>0)
            cout<<"The Number of occurance of the String:"
                    << count<<endl;
            else
            cout<<"count="<<count<<"The Sub sstring in the given
                string  is not found...!!"<<endl;
        }
        void replace()
        {
            int count=0;
            char chr,rchr;
            cout<<"Enter the Character that is to be replaced";
            cin>>chr;
            cout<<"Enter the Character for replacing.....";
            cin>>rchr;
            for(int i=0;i<strlen(str);i++)
            {
                if(str[i]==chr)
                {
                    str[i]=rchr;
                    count=count+1;
                }
            }
            if(count>0)
            cout<<"Modified String is:"<<str<<endl;
            else
            cout<<chr<<"is not present in the string"<<str<<endl;
        }
    };
    void main()
    {
        String s1;
        int ch;
        char sub1[100];
        while(1)
        {
                cout<<"1.To Concatenate the string\t2.To find the
Occurance\t3.To replace\n";
                cout<<"Enter the choice";
                cin>>ch;
                switch(ch)
                {
                        case 1:
                        {
                                cout<<"Enter the string to concatenate:";
                                cin>>sub1;
                                String s2(sub1);
                                String s3(s1,s2);
                        }
                                break;
                        case 2:
                        {
                                s1.occuarance();
                        }
                        break;
                        case 3:
                        {
                                s1.replace();
                        }
                        break;
                        default: cout<<"Enter the vaild choice";
                                 exit(0);
                }
        }
        getch();
}
Run:
```



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ g++ pr9.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ ./a.out
Enter the String:
Proogramming
1.To Concatenate the string     2.To find the Occurance 3.To replace
Enter the choice1
Enter the string to concatenate:c
AFTER CONCATENATION :Proogrammingc
1.To Concatenate the string     2.To find the Occurance 3.To replace
Enter the choice2
Enter  the Sub String:mm
The Number of occurance of the String:1
1.To Concatenate the string     2.To find the Occurance 3.To replace
Enter the choice3
Enter the Character that is to be replacedo
Enter the Character for replacing.....r
Modified String is:Prrrgrammingc
1.To Concatenate the string     2.To find the Occurance 3.To replace
Enter the choice4
Enter the valid choiceamoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ 
```

10. Consider an example of a bookshop which sells books and video tapes. These two classes are inherited from the base class called ***media.*** The media class has data members such as ***title* and *publication*.** The ***book*** class has data members for storing a number of pages in a book, and the ***tape*** class has the playing time in a tape. Each class will have member functions such as ***read( )*** and ***show( ).*** In the base class, these members have to be defined as virtual functions. Write a program which models the class hierarchy for the bookshop and processes objects of these classes using pointers to the base class.

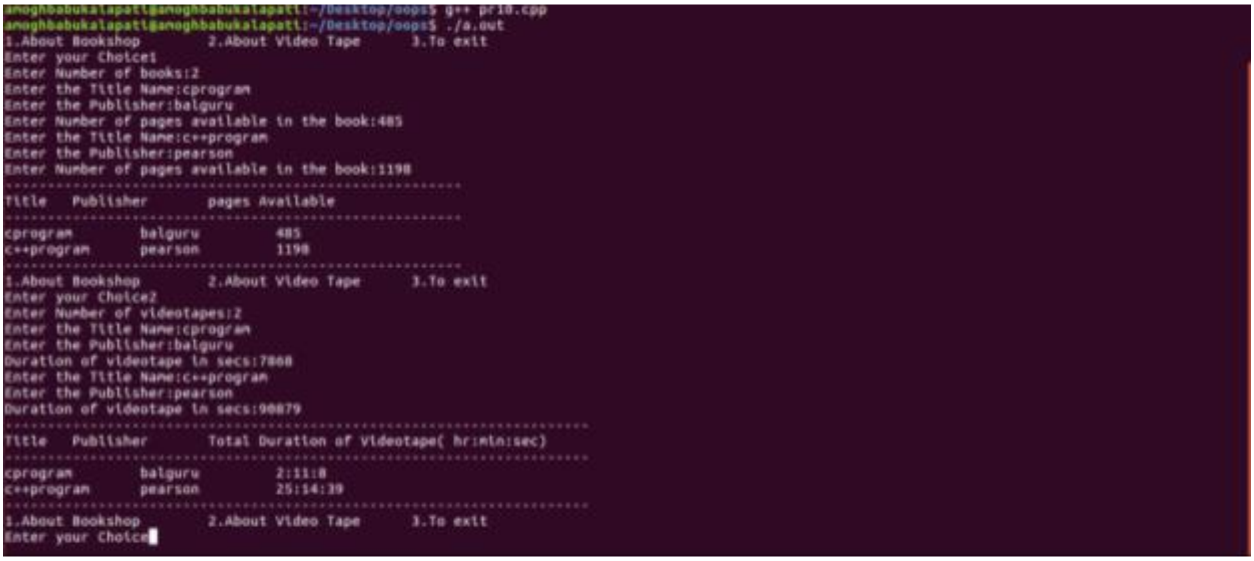
Program:

```
#include<iostream.h>
#include<stdlib.h>
#include<string.h>
class media
{
        protected:
        char title[100],pub[100];
        public:
        virtual void read()
        {
                cout<<"Enter the Title Name:";
                cin>>title;
                cout<<"Enter the Publisher:";
                cin>>pub;
        }
        virtual void show()
        {
                cout<<title<<"\t"<<pub;
```

```
	}
};
class books:public media
{
	int page;
	public:
	void read()
	{
		media::read();
		cout<<"Enter Number of pages available in the book:";
		cin>>page;
	}
	void show()
	{
		media::show();
		cout<<"\t\t"<<page<<endl;
	}
};
class videotape:public media
{
	int hr,min,sec;
	public:
	void read()
	{
		media::read();
		cout<<"Duration of videotape in secs:";
		cin>>sec;
		compute();
	}
	void compute()
	{
		min=sec/60;
		sec=sec%60;
		hr=min/60;
		min=min%60;
	}
```

```
        void show()
        {
                media::show();
                cout<<"\t\t"<<hr<<":"<<min<<":"<<sec<<endl;
        }
};
void main()
{
        media *m;
        books bs[50];
        videotape vt[50];
        int ch,nb,nv;
        while(1)
        {
                cout<<"1.About Bookshop\t2.About Video Tape\t3.To
exit"<<endl;
                cout<<"Enter your Choice";
                cin>>ch;
                switch(ch)
                {
                        case 1: cout<<"Enter Number of books:";
                                cin>>nb;
                                for(int i=1;i<=nb;i++)
                                {
                                        m=&bs[i];
                                        m->read();
                                }
                cout<<"----------------------------------------------------"<<endl;
                                cout<<"Title\t"<<"Publisher\t"<<"pages
Available"<<endl;
                                cout<<"----------------------------------------------------
"<<endl;
                                for(int i=1;i<=nb;i++)
                                {
                                        m=&bs[i];
                                        m->show();
                                }
                                cout<<"----------------------------------------------------------
"<<endl;
                                break;
                        case 2: cout<<"Enter Number of videotapes:";
                                cin>>nv;
                                for(int i=1;i<=nv;i++)
                                {
                                        m=&vt[i];
                                        m->read();
                                }
cout<<"---------------------------------------------------------------------------"<<endl;
                                cout<<"Title\t"<<"Publisher\t"<<"Total Duration of
Videotape( hr:min:sec)"<<endl;
cout<<"---------------------------------------------------------------------------"<<endl;
                                for(int i=1;i<=nv;i++)
                                {
                                        m=&vt[i];
                                        m->show();
                                }
cout<<"---------------------------------------------------------------------------"<<endl;
                                break;
                        default:cout<<"Enter the valid choice....!!!!!!your are
terminated now";
                                exit(0);
                }
        }
        getch(0);
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ g++ pr10.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops$ ./a.out
1.About Bookshop        2.About Video Tape      3.To exit
Enter your Choice1
Enter Number of books:2
Enter the Title Name:cprogram
Enter the Publisher:balguru
Enter Number of pages available in the book:485
Enter the Title Name:c++program
Enter the Publisher:pearson
Enter Number of pages available in the book:1198
-----------------------------------------------------
Title   Publisher       pages Available
-----------------------------------------------------
cprogram        balguru         485
c++program      pearson         1198
-----------------------------------------------------
1.About Bookshop        2.About Video Tape      3.To exit
Enter your Choice2
Enter Number of videotapes:2
Enter the Title Name:cprogram
Enter the Publisher:balguru
Duration of videotape in secs:7868
Enter the Title Name:c++program
Enter the Publisher:pearson
Duration of videotape in secs:90879
--------------------------------------------------------------------
Title   Publisher       Total Duration of Videotape( hr:min:sec)
--------------------------------------------------------------------
cprogram        balguru         2:11:8
c++program      pearson         25:14:39
--------------------------------------------------------------------
1.About Bookshop        2.About Video Tape      3.To exit
Enter your Choice
```

11.Write a C++ Program to create class STUDENT ,with data members USN, name and age. Using Inheritance , create classes UGSTUDENT and PGSTUDENT having fields as semester, fees and stipend. Enger the data for atleast 5 students .Find the Semester wise average age for all UG and PG Students separately .

Program:

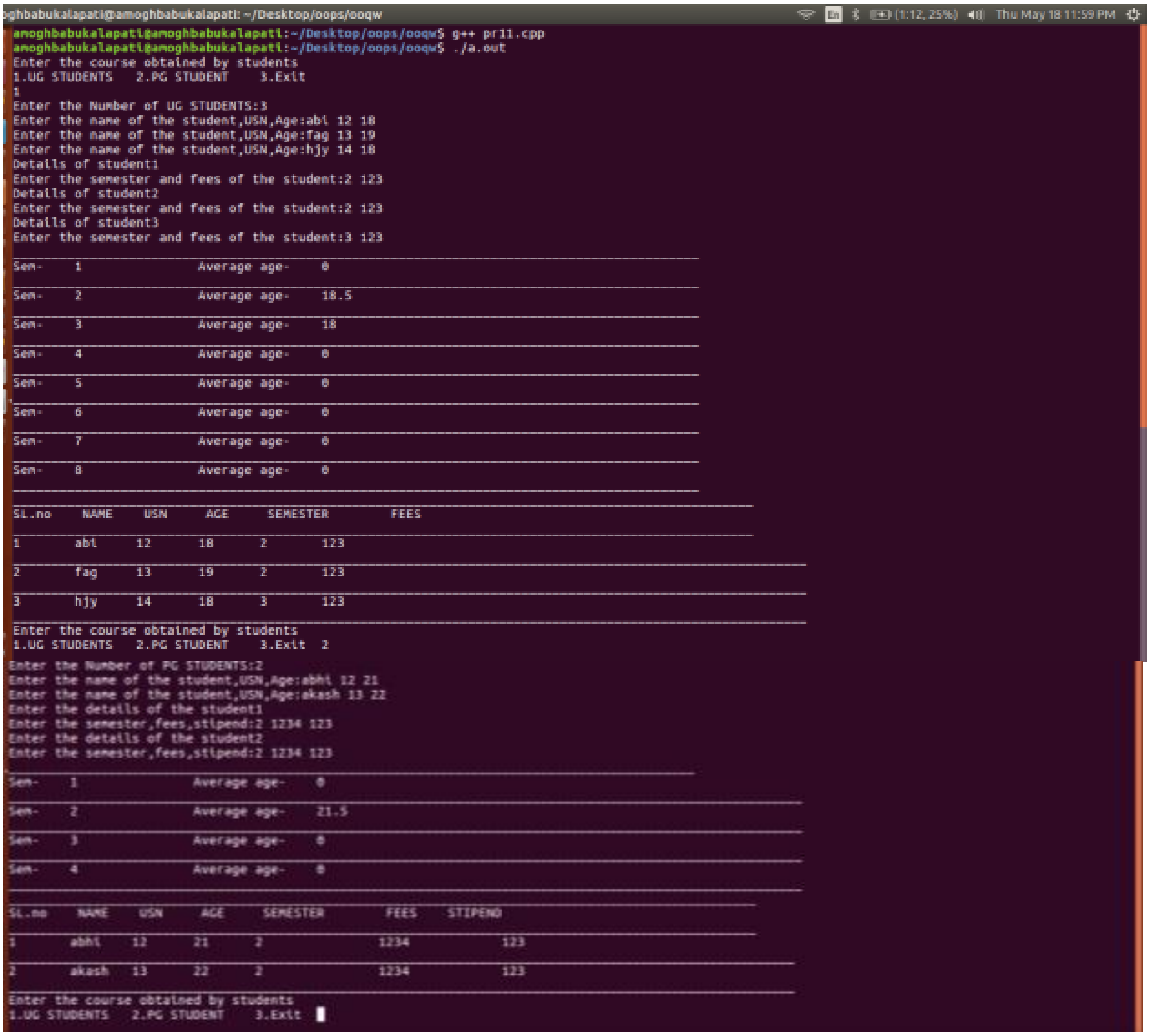
```
#include<iostream.h>
#include<stdlib.h>
class student
{
        protected:
        struct stu
        {
                char name[50],usn[50];
                int fees,age,sem1,sem2,stipend;
        }s[50];
             public:
        void getdata(int n)
        {
                for(int i=1;i<=n;i++)
                {
                        cout<<"Enter the name of the student,USN,Age:";
                        cin>>s[i].name>>s[i].usn>>s[i].age;
                }
        }
};
class ugstudent:public student
{
        protected:
        int count1;
        float sum1;
        public:
        void ugdata(int n)
        {
                for(int i=1;i<=n;i++)
                {
                        cout<<"Details of student"<<i<<"\n";
                        cout<<"Enter the semester and fees of the student:";
                        cin>>s[i].sem1>>s[i].fees;
                }
        }
        void putug(int n)
        {
cout<<"_______________________________________________
_______________________________________\n";
                cout<<"SL.no \t NAME \t USN \t AGE \t SEMESTER \t FEES
\n";
cout<<"_______________________________________________
_______________________________________\n";
                for(int i=1;i<=n;i++)
                {
cout<<i<<"\t"<<s[i].name<<"\t"<<s[i].usn<<"\t"<<s[i].age<<"\t"<<s[i].se
m1<<"\t"<<s[i].fees<<"\n";
cout<<"_______________________________________________
_____________________________________________\n";
                }
        }
        void computeug(int n)
        {
                float total=0.0;
```

```
		sum1=0.0,count1=0;
cout<<"__________________________________________________________
_______________________________________\n";
		for(int i=1;i<=8;i++)
		{
			for(int j=1;j<=n;j++)
			{
				if(i==s[j].sem1)
				{
					sum1=sum1+s[j].age;
					count1++;
				}
			}
			if(count1!=0)
			total=(sum1/count1);
			else
			total=0.0;
			cout<<"Sem-"<<"\t"<<i<<"\t\t"<<"Average age-
"<<"\t"<<total<<"\n";
cout<<"__________________________________________________________
_______________________________________\n";
			sum1=0.0,count1=0;
		}
	}
};
class pgstudent:public student
{
	protected:
	int count2;
	float sum2,total;
	public:
	void pgdata(int n)
	{
		for(int i=1;i<=n;i++)
		{
			cout<<"Enter the details of the student"<<i<<"\n";
```

```
            cout<<"Enter the semester,fees,stipend:";
            cin>>s[i].sem2>>s[i].fees>>s[i].stipend;
        }
    }
    void putpg(int n)
    {
cout<<"____________________________________________________________
__________________________________________________\n";
        cout<<"SL.no \t NAME \t USN \t AGE \t SEMESTER \t FEES
\t STIPEND\n";
cout<<"____________________________________________________________
__________________________________________________\n";
        for(int i=1;i<=n;i++)
        {
cout<<i<<"\t"<<s[i].name<<"\t"<<s[i].usn<<"\t"<<s[i].age<<"\t"<<s[i].se
m2<<"\t\t"<<s[i].fees<<"\t\t"<<s[i].stipend<<endl;
cout<<"____________________________________________________________
________________________________________________________\n";
        }
    }
    void computepg(int n)
    {
        float total=0.0;
        sum2=0.0,count2=0;
cout<<"____________________________________________________________
_________________________________________\n";
        for(int i=1;i<=4;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if(i==s[j].sem2)
                {
                    sum2=sum2+s[j].age;
                    count2++;
                }
            }
```

```
                        if(count2!=0)
                        total=sum2/count2;
                        else
                        total=0.0;
                        cout<<"Sem-"<<"\t"<<i<<"\t\t"<<"Average age-
"<<"\t"<<total<<"\n";
cout<<"____________________________________________________________
________________________________________________________\n";
                        sum2=0.0,count2=0;
                }
        }
};
void main()
{
        int ch,uc1,uc2;
        ugstudent U;
        pgstudent P;
        while(1)
        {
        cout<<"Enter the course obtained by students\n";
        cout<<"1.UG STUDENTS\t2.PG STUDENT\t3.Exit\t";
        cin>>ch;
                        switch(ch)
                          {
                                        case 1: cout<<"Enter the Number of UG
STUDENTS:";
                                                cin>>uc1;
                                                U.getdata(uc1);
                                                U.ugdata(uc1);
                                                 U.computeug(uc1);
                                                 U.putug(uc1);
                                                break;
                                        case 2: cout<<"Enter the Number of PG
STUDENTS:";
                                                cin>>uc2;
                                                P.getdata(uc2);
                                        P.pgdata(uc2);
                                         P.computepg(uc2);
                                         P.putpg(uc2);
                                        break;
                                   case 3:exit(0);
                         }
              }
              getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ g++ pr11.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ ./a.out
Enter the course obtained by students
1.UG STUDENTS   2.PG STUDENT    3.Exit
1
Enter the Number of UG STUDENTS:3
Enter the name of the student,USN,Age:abl 12 18
Enter the name of the student,USN,Age:fag 13 19
Enter the name of the student,USN,Age:hjy 14 18
Details of student1
Enter the semester and fees of the student:2 123
Details of student2
Enter the semester and fees of the student:2 123
Details of student3
Enter the semester and fees of the student:3 123
------------------------------------------------------------------------------------------
Sem-    1               Average age-    0
------------------------------------------------------------------------------------------
Sem-    2               Average age-    18.5
------------------------------------------------------------------------------------------
Sem-    3               Average age-    18
------------------------------------------------------------------------------------------
Sem-    4               Average age-    0
------------------------------------------------------------------------------------------
Sem-    5               Average age-    0
------------------------------------------------------------------------------------------
Sem-    6               Average age-    0
------------------------------------------------------------------------------------------
Sem-    7               Average age-    0
------------------------------------------------------------------------------------------
Sem-    8               Average age-    0
------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
SL.no    NAME    USN     AGE     SEMESTER        FEES
-------------------------------------------------------------------------------------------------
1       abl     12      18      2       123
--------------------------------------------------------------------------------------------------------
2       fag     13      19      2       123
--------------------------------------------------------------------------------------------------------
3       hjy     14      18      3       123
--------------------------------------------------------------------------------------------------------
Enter the course obtained by students
1.UG STUDENTS   2.PG STUDENT    3.Exit  2
Enter the Number of PG STUDENTS:2
Enter the name of the student,USN,Age:abhi 12 21
Enter the name of the student,USN,Age:akash 13 22
Enter the details of the student1
Enter the semester,fees,stipend:2 1234 123
Enter the details of the student2
Enter the semester,fees,stipend:2 1234 123
------------------------------------------------------------------------------------------
Sem-    1               Average age-    0
--------------------------------------------------------------------------------------------------------
Sem-    2               Average age-    21.5
--------------------------------------------------------------------------------------------------------
Sem-    3               Average age-    0
--------------------------------------------------------------------------------------------------------
Sem-    4               Average age-    0
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
SL.no    NAME    USN     AGE     SEMESTER        FEES    STIPEND
--------------------------------------------------------------------------------------------------
1       abhi    12      21      2               1234            123
--------------------------------------------------------------------------------------------------------
2       akash   13      22      2               1234            123
--------------------------------------------------------------------------------------------------------
Enter the course obtained by students
1.UG STUDENTS   2.PG STUDENT    3.Exit
```

12.Create a class ‘**Shape**’. It should have no data members but have a pure virtual function **get_area().**

Derive ‘**Rectangle**’ and ’**Ellipse**’ classes from the class ’**Shape**’. Both should have two data members. **Rectangle** class data members are width and height, ’**Ellipse**’ class data members are minor and major axis. Override the **’shape :: get_area( )’** function inside both the derived classes which returns the area . Also, write a constructor for these derived classes.

Create a class ’**Canvas**’. It should have no data members. Its only member function as **’display( )’** with reference of class ‘**Shape**’ as a formal argument. With this reference, call the **‘Shape::get _area( )’** function inside **’Canvas::display( )’** function.

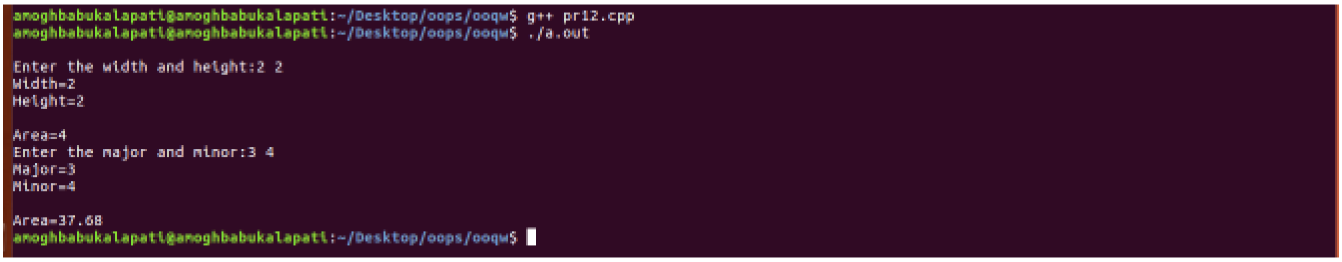
Write a **‘main( )’** function to utilize these classes. Declare objects of classes ’**Rectangle**’, **‘Ellipse’, and ‘Canvas’**. Call the **‘Canvas::display()’** function first by passing the object of class ‘**Rectangle’** and then by passing the object of class **‘Ellipse’** to it .

Program:

```
#include<iostream.h>
#include<stdlib.h>
class shape
{
        public:
        virtual float getarea()=0;
};
class rectangle:public shape
{
        float width,height;
        public:
        rectangle()
        {
        }
        float getarea()
        {
                float r_area;
                cout<<endl<<"Enter the width and height:";
        cin>>width>>height;
        r_area=width*height;
        cout<<"Width="<<width<<endl<<"Height="<<height<<endl;
        return(r_area);
    }
};
class ellipse:public shape
{
    float major,minor;
    public:
    ellipse()
    {
    }
    float getarea()
    {
        float e_area;
        cout<<endl<<"Enter the major and minor:";
        cin>>major>>minor;
        e_area=major*minor*3.14;
        cout<<"Major="<<major<<endl<<"Minor="<<minor<<endl;
        return(e_area);
    }
};
class canvas
{
    float f;
    public:
    void display(shape &sh)
    {
        f=sh.getarea();
        cout<<endl<<"Area="<<f;
    }
};
void main()
{
    rectangle r;
```

```
        ellipse e;
        canvas c;
        c.display(r);
        c.display(e);
        cout<<endl;
        getch();
}
```

Run:



```
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ g++ pr12.cpp
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ ./a.out

Enter the width and height:2 2
Width=2
Height=2

Area=4
Enter the major and minor:3 4
Major=3
Minor=4

Area=37.68
amoghbabukalapati@amoghbabukalapati:~/Desktop/oops/ooqw$ 
```