Experiment No: 1

Write a program to find the frequency of presence an element in an array.

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
class  frequency
{
     private:
               int  n, m[100], ele, freq;
     public:
               void getdata();
               void findfreq();
               void display();
};
void  frequency::getdata()
{
     cout<<"Enter the size of the array: ";
     cin>>n;
     cout<<"Enter "<<n<<" elements into the array:   ";
     for(int i=0; i<n; i++)
          cin>>m[i];
     cout<<"Enter the search element:   ";
     cin>>ele;
}
void  frequency::findfreq()
{
     freq = 0;
     for(int i=0; i<n; i++)
          if(ele==m[i])
               freq++;
}
void  frequency::display()
{
     if(freq > 0)
          cout<<"Frequency of "<<ele<<" is "<<freq;
     else
          cout<<ele<<" does not exist";
}
```

```
void  main()
{
     frequency F;
     clrscr();
     F.getdata();
     F.findfreq();
     F.display();
     getch();
}
```

```
Enter the size of the array: 5
Enter 5 elements into the array:  10 50 40 30 40
Enter the search element:  40
Frequency of 40 is 2
```

```
Enter the size of the array: 5
Enter 5 elements into the array:  10 20 50 40 30
Enter the search element:  25
25 does not exist
```

RUN

RUN

Experiment No: 2

Write a program to insert an element into an array at a given position.

```
#include<iostream.h>
#include<iomanip.h>
class insertion
{
     private: int n, m[100], ele, p;
     public:
               void getdata();
               void insert();
               void display();
};
void insertion::getdata()
{
     cout<<"How many elements? ";
     cin>>n;
     cout<<"Enter the elements: ";
     for(int i=0; i<n; i++)
          cin>>m[i];
     cout<<"Enter the element to be inserted: ";
     cin>>ele;
     cout<<"Enter the position ( 0 to "<<n<<"): ";
     cin>>p;
}
void insertion::insert()
{
     if(p > n)
     {
          cout<<p<<" is an invalid position";
          exit(0);
     }
     for(int i=n-1; i>=p; i--)
          m[i+1] = m[i];
     m[p] = ele;
     n++;
     cout<<ele<<" is successfully inserted"<<endl;
}
```

```
void  insertion::display()
{
     cout<<"The array after the insertion is ";
     for(int i=0; i<n; i++)
          cout<<setw(4)<<m[i];
}
void  main()
{
     insertion I;
     I.getdata();
     I.insert();
     I.display();
}
```

```
How many elements? 5
Enter the elements: 20 30 40 50 60
Enter the element to be inserted: 10
Enter the position ( 0 to 5): 0
10 is successfully inserted into position 0
The array after the insertion is   10   20   30   40   50   60
```

```
How many elements? 5
Enter the elements: 10 20 30 50 60
Enter the element to be inserted: 40
Enter the position ( 0 to 5): 3
40 is successfully inserted into position 3
The array after the insertion is   10   20   30   40   50   60
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the element to be inserted: 60
Enter the position ( 0 to 5): 5
The array after the insertion is   10   20   30   40   50   60
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the element to be inserted: 60
Enter the position ( 0 to 5): 7
7 is an invalid position
```

Experiment No: 3

Write a program to delete an element from an array from a given position.

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
class deletion
{
     private:
          int m[100], n, ele, p;
     public:
          void getdata();
          void remove();
          void display();
};
void deletion::getdata()
{
     cout<<"How many elements? ";
     cin>>n;
     cout<<"Enter the elements: ";
     for(int i=0; i<n; i++)
          cin>>m[i];
     cout<<"Enter the position (0 to "<<n-1<<"): ";
     cin>>p;
}
void deletion::remove()
{
     if(p > n-1)
     {
          cout<<p<<" is an invalid position";
          exit(0);
     }
     ele = m[p];
     for(int i=p+1; i<n; i++)
          m[i-1] = m[i];
     n--;
     cout<<ele<<" is successfully removed"<<endl;
}
```

```
void  deletion::display()
{
     cout<<"The array after deletion is ";
     for(int i=0; i<n; i++)
          cout<<setw(4)<<m[i];
}
void  main()
{
     deletion  D;
     clrscr();
     D.getdata();
     D.remove();
     D.display();
     getch();
}
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the position (0 to 4): 0
The element 10 at position 0 is successfully removed
The array after the deletion is    20   30   40   50
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the position (0 to 4): 3
The element 40 at position 3 is successfully removed
The array after deletion is    10   20   30   50
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the position (0 to 4): 4
The element 50 at position 4 is successfully removed
The array after the deletion is    10   20   30   40
```

```
How many elements? 5
Enter the elements: 10 20 30 40 50
Enter the position (0 to 4): 5
5 is an invalid position
```

Experiment No: 4

Write a program to sort the elements of an array in ascending order using insertion sort.

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
class  sorting
{
     private:
          int m[100], n;
     public:
          void getdata();
          void sort();
          void display();
};
void  sorting::getdata()
{
     cout<<"How many elements? ";
     cin>>n;
     cout<<"Enter the elements: ";
     for(int i=0; i<n; i++)
          cin>>m[i];
}
void  sorting::display()
{
     cout<<"The array after sorting is ";
     for(int i=0; i<n; i++)
          cout<<setw(4)<<m[i];
}
void  sorting::sort()
{
     int temp, j;
     for(int i=1; i<n; i++)  //i = the pass number
     {
          j = i;              //j = the comparison position
          while(j>=1)
          {
               if(m[j]<m[j-1])    //m[j]=comparison element
```

```
                {
                     temp = m[j];
                     m[j] = m[j-1];
                     m[j-1] = temp;
                }
                j--;
           }
      }
}
void main()
{
     sorting  S;
     clrscr();
     S.getdata();
     S.sort();
     S.display();
     getch();
}
```

```
Enter the size of the array: 5
Enter the elements: 50 30 40 20 10
Elements after sorting: 10   20   30   40   50
```

RUN

RUN

Experiment No: 5

Write a program to search for a given element in an array using Binary search method.

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
class binarysearch
{
     private:
          int m[100], n, ele, loc;
     public:
          void getdata();
          void search();
          void display();
};
void binarysearch::getdata()
{
     cout<<"How many elements? ";
     cin>>n;
     cout<<"Enter the elements: ";
     for(int i=0; i<n; i++)
          cin>>m[i];
     cout<<"Enter the search element:  ";
     cin>>ele;

}
void binarysearch::display()
{
     if(loc >= 0)
          cout<<"Position= "<<loc;
     else
          cout<<"Search is unsuccessful";
}
void binarysearch::search()
{
     int beg, end, mid;
     loc = -1;
     beg = 0;
     end = n-1;
```

```
        while(beg <= end)
        {
             mid = (beg+end)/2;
             if(ele == m[mid])
             {
                  loc = mid;
                  break;
             }
             else
                  if(ele < m[mid])
                       end = mid-1;
                  else
                       beg = mid + 1;
        }
   }
   void main()
   {
        binarysearch  B;
        clrscr();
        B.getdata();
        B.search();
        B.display();
        getch();
   }
```

```
How many elements? 5
Enter the elements: 10 40 50 30 50
Enter the search element:  50
Position= 2
```

```
How many elements? 5
Enter the elements: 40 50 20 30 10
Enter the search element:  60
Search is unsuccessful
```

RUN

Experiment No: 6

Write a program to create a class with data members principle, time and rate. Create member functions to accept data values to compute simple interest and to display the result.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class  interest
{
     private:
          double p, t, r, si;
     public:
          void  getdata();
          void  compute();
          void  putdata();
};
void  interest::getdata()
{
     cout<<"Enter principle amount, time and rate"<<endl;
     cin>>p>>t>>r;
}
void  interest::putdata()
{
     cout<<"Principle:   "<<p<<endl;
     cout<<"Time:   "<<t<<endl;
     cout<<"Rate:   "<<r<<endl;
     cout<<"Simple interest:   "<<si<<endl;
}
void  interest::compute()
{
      si = (p*t*r)/100;
}
void   main()
{
     interest  I;
     clrscr();
     I.getdata();
     I.compute();
```

```
        I.putdata();
    }
```

```
Enter principle amount, time and rate:    5000 5.5 8.75
Principle:   5000
Time:  5.5
Rate:   8.75
simple  interest:  2406.25
```

RUN

RUN

RUN