1. A university wants to automate their admission process. Students are admitted based on marks
   scored in a qualifying exam. A student is identified by student id, age and marks in qualifying exam. Data are valid, if:

   - Age is greater than 20

   - Marks is between 0 and 100 (both inclusive)
     A student qualifies for admission, if

```
#include<iostream>

using namespace std;
class student
{
   int age,marks;
   string id;
   public:
      void setdata();
      void check_qualification();
      void display();
};

main()
{
   student stu;
   stu.setdata();
   stu.check_qualification();
}

void student::setdata()
{
   cout<<"Enter student ID:";
   cin>>id;
   cout<<"Enter the age:";
   cin>>age;
   cout<<"Enter the marks:";
   cin>>marks;

}

void student::check_qualification()
{
   if (age>20)
   {
      if ((marks<=100) && (marks>=0))
      {
         if (marks>=65) display();
         else cout<<"Disqualified";
      }
      else cout<<"Invalid data";
   }
   else
      cout<<"invalid data";
}
```

```
void student::display()
{
   cout<<"Student ID:"<<id<<'\n';
   cout<<"Age:"<<age<<'\n';
   cout<<"Marks"<<marks<<"\n";
}
```

7. Write a C++ program to add two distances using passing object as arguments and returning objects.

```
#include <bits/stdc++.h>

using namespace std;

class Example {

public:

   int a;



   // This function will take

   // an object as an argument

   void add(Example E)

   {

         a = a + E.a;

   }

};

// Driver Code
```

```
int main()
{

  // Create objects
  Example E1, E2;

  // Values are initialized for both objects
  E1.a = 50;
  E2.a = 100;

  cout << "Initial Values \n";
  cout << "Value of object 1: " << E1.a
       << "\n& object 2: " << E2.a
       << "\n\n";
  E2.add(E1);
  // Changed values after passing
  // object as argument
  cout << "New values \n";
  cout << "Value of object 1: " << E1.a
       << "\n& object 2: " << E2.a
       << "\n\n";
  return 0;
}
```

7. Write a program to implement bubble sort for an array.

```
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int n)
{
  int i, j;
  for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - i - 1; j++)
              if (arr[j] > arr[j + 1])
                    swap(arr[j], arr[j + 1]);
}
void printArray(int arr[], int size)
{
  int i;
  for (i = 0; i < size; i++)
        cout << arr[i] << " ";
  cout << endl;
}
main()
{
  int arr[] = { 5, 1, 4, 2, 8};
  int N = sizeof(arr) / sizeof(arr[0]);
```

```
    bubbleSort(arr, N);
    cout << "Sorted array: \n";
    printArray(arr, N);
    return 0;
}
```

7. Write a program to find max of two numbers and max of three numbers using function overloading.

```
#include<iostream>
using namespace std;
class findmax
{
   private:
      int k,l,m;
   public:
      void compare(int a,int b)
      {
         if(a>b)
            cout<<a<<" is bigger than "<<b;
         else
            cout<<b<<" is bigger than "<<a;
      }
      void compare(float a,float b,float c)
```

```
        {
            if((a>=b) &&(a>=c))
                cout<<a<<" is max";
            else if ((b>=c)&&(b>=a))
            {
                cout<<b<<" is max";
            }
            else if((c>=a)&&(c>=b))
            {
                cout<<c<<" is max";
            }
        }
};
main()
{
    findmax find;
    int a,b,c,k;
    cout<<"enter 1 to comapre 2 numbers \n enter 2 to comapre three numbers";
    cin>>k;
    switch (k)
    {
    case 1:
        cout<<"enter two number with space";
```

```
        cin>>a>>b;
        find.compare(a,b);
        break;
    case 2:
        cout<<"enter three number with space";
        cin>>a>>b>>c;
        find.compare(float(a),float(b),float(c));
        break;
    default:
        break;
    }
}
```

7. Implement a program to swap two strings if the given strings are not equal using call by value,
   call by reference.

```
#include<iostream>
using namespace std;
void swap(string &a,string &b)
{
    cout<<"before swap :\n";
    cout<<"string A ="<<a<<"    string B ="<<b;
    string k;
```

```
    k=a;
    a=b;
    b=a;
    cout<<"after swap :\n";
    cout<<"string A ="<<a<<"    string B ="<<b;
}
main()
{
    string *k;
    string *l;
    cout<<"enter strings with space :";
    cin>>*k>>*l;
    swap(*k,*l);


}
```

7. Create a student class with the private attributes as rollno, name, maths_mark, phy_mark and chem_mark. Implement the following in different methods:
    1. Compute_cutoff ( ) - find the cutoff marks of a student,
    2. Print( ) - Display student details with their cutoff mar

```
#include<iostream>
using namespace std;
```

```
class student_info
{
    private:
        int roll,math_mark,phy_mark,chem_mark;
        char name[10];
    public:
        void details();
        float cutoff();
        void getdetails();

};

main()
{
    int N;
    cout<<"Enter total number of students:";
    cin>>N;
    student_info info[N];
    for (int i = 0; i < N; i++)
    {
        cout<<"enter details of student"<<i+1<<": "<<"\n";
        info[i].getdetails();
```

```
    }
    for (int i = 0; i <= N; i++)
    {
        info[i].details();
    }
}

float student_info::cutoff()
{
    float cut;
    cut=math_mark+(phy_mark+chem_mark)/2;
    return cut;
}

void student_info::details()
{
    cout<<"name :"<<name<<"\n";
    cout<<"roll number ="<<roll<<"\n";
    float k=cutoff();
    cout<<"cutoff ="<<k<<"\n";

}
```

```
void student_info::getdetails()
{
    cout<<"enter roll number :";
    cin>>roll;
    cout<<"enter name ";
    cin>>name;
    cout<<"enter maths mark :";
    cin>>math_mark;
    cout<<"enter chem mark :";
    cin>>chem_mark;
    cout<<"enter phy mark :";
    cin>>phy_mark;

}
```