Write a program that can convert the Distance (meter, centimeter) to meters measurement in float and vice versa. Make a class distance with two data members, meter and centimeter. You can add function members as per your requirement.

```
#include <iostream>
#define SUCCESS 0
using namespace std;
class Distance{
  int meter;
  int centimeter;
public:
  Distance(int m,int cm):meter(m),centimeter(cm){};
  Distance(float no)
  {
    meter=no;
    centimeter=(no-int(no))*100;
  }
  operator float()
  {
    return float(meter) + float(centimeter)/100.0;
  }
  Distance operator+(Distance a)
  {
    return
Distance(meter+a.meter+(centimeter+a.centimeter)/100,(centimeter+a.centimeter)%
100);
  }
  void display()
  {
    cout << meter << " m " << centimeter << "cm" << endl;
  }
};
int main()
{
  Distance d(10.34);
  d.display();
  cout << "Distance typecasting to float ";
  cout << d << endl;
  Distance c(10,6);
  Distance s = d+c;
```

```
 cout << "addition of two distances" << endl;
 d.display();
 c.display();
 cout << "Gives";
 s.display();
 return SUCCESS;
}

#include<iostream>//or
using namespace std;
class distanc
{
   float meter ,centimeter;
public:
   distanc(float m,float cm)
   {
      meter=m;
      centimeter=cm;
   }
   distanc(float m)
   {
      meter=m;
      centimeter=0;
   }
   operator float()
   {
      return (meter+(centimeter/100));
   }
   void display()
   {
      int m;
      float c;
      m=static_cast<int>(meter);
      c=(meter-m)*100;
      cout<<m<<" meter , "<<c<<" centimeter"<<endl;
   }
};
int main()
{
   distanc d1(2.1,56.5),d2(.563);
```

```
   float disp;
   d2.display();
   disp=d1;
   cout<<"meter = "<<d1;
}
```