Unary Operator Overloading - class Distance (Id-13671)

The program must accept a **distance** in **feet** and **inches**. The program must add **1 inch** to the given distance and print the revised distance. Then the program must add **1 inch** to the revised distance and print the revised distance.

Your task is to define the class **Distance** so that the program runs successfully.

Example Input/Output 1:
Input:
2 9

Output:
2 10
2 11

Example Input/Output 2:
Input:
4 11

Output:
5 0
5 1

Show My Solution

2212046@nec 11-Sep-2023 14:08:01 DINESHRAM A 2212046 CSE

```
#include <iostream>
using namespace std;
class Distance{
    public:
    int f,i;
    Distance(int a,int b){
        f=a;
        i=b;
    }
    friend Distance operator ++(Distance &a){
        a.i++;
        if(a.i>=12){
            a.f++;
            a.i=0;
        }
    }void display(){
        cout<<f<<" "<<i<<endl;
    }
};
int main()
{
    int feet, inches;
    cin >> feet >> inches;
    Distance distance(feet, inches);
    ++distance;
    distance.display();
    ++distance;
    distance.display();
    return 0;
}
```

Marks of Two Students (Id-13858)

The program must accept the marks of two students as the input and print their sum as the output. Please fill in the missing lines of code by defining the class **Student** so that the program runs successfully.

Input Format:
The first line contains two integers representing the marks of two students separated by a space.

Output Format:
The first line contains an integer representing the sum of the marks of the two students.

Example Input/Output 1:
Input:
45 80

Output:
125

Explanation:
The marks of the first student is **45**.
The marks of the second student is **80**.
So their sum is **125** which is printed as the output.

Example Input/Output 2:
Input:
75 99

Output:
174

Show My Solution

2212046@nec | 11-Sep-2023 14:37:24 | DINESHRAM A | 2212046 | CSE

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

class Student
{
   public:
   int a;
   void setMarks(int i){
       a=i;
   }
   Student operator +(Student &stud2){
       Student s3;
       s3.a=a+stud2.a;
       return s3;
   }
    int getMarks(){
       return a;
   }
};

int main()
{
    int m1, m2;
    cin >> m1 >> m2;
    Student stud1, stud2;
    stud1.setMarks(m1);
    stud2.setMarks(m2);
    Student stud3 = stud1 + stud2;
    cout << stud3.getMarks();
    return 0;
}
```

Sum of Two Distances (Id-13859)

The program must accept two distances(in feet and inches) as the input. The program must print the sum of the two distances(in feet and inches) as the output. Please fill in the missing lines of code by defining the class **Distance** so that the program runs successfully.

Formula: 1 Foot = 12 Inches

Input Format:
The first line contains two integers representing the feet and inches of the distance 1.
The second line contains two integers representing the feet and inches of the distance 2.

Output Format:
The first line contains two integers representing the feet and inches of the sum of the two distances.

Example Input/Output 1:
Input:
22 10
23 11

Output:
46 9

Explanation:
Distance 1 = **22 feet** and **10 inches**.
Distance 2 = **23 feet** and **11 inches**.
The sum of the two distances is **46 feet** and **9 inches**.

Example Input/Output 2:
Input:
15 8
11 4

Output:
27 0

Show My Solution

2212046@nec | 11-Sep-2023 14:47:35 | DINESHRAM A | 2212046 | CSE

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

class Distance
{
  public:
  int x,y;
  void setFeet(int i){
      x=i;
  }
  void setInches(int j){
      y=j;
  }
  Distance operator +(Distance &d2){
      Distance d3;
      d3.x=x+d2.x;
      d3.y=y+d2.y;
      return d3;
  }
  int getFeet(){
      if(y>=12){
          return x+1;
      }return x;
  }
  int getInches(){
      if(y>=12){
          return y-12;
      }return y;
  }
};

int main()
{
    int feet, inches;
    cin >> feet >> inches;
    Distance d1;
    d1.setFeet(feet);
    d1.setInches(inches);
    cin >> feet >> inches;
    Distance d2;
    d2.setFeet(feet);
    d2.setInches(inches);
    Distance d3 = d1 + d2;
    cout << d3.getFeet() << " " << d3.getInches();
    return 0;
}
```