In [1]:
```
A, B, A1, A2, B1, B2 = input("Enter values: ").split()
if (A==A1 and B==A2):
    print(1)
else:
    if(A==B1 and B==B2):
        print(2)
    else:
        print(0)

if (A==A1 and B==A2):
    print(1)
else:
    if(A==B1 and B==B2):
        print(2)
    else:
        print(0)
```

```
Enter values: 2 4 1 2 4 1
0
0
```

In [2]:
```
a1, a2, a3, a4 = map(int, input().split())
cnt=0
if(a1!=a2):
    cnt=cnt+1
    if(a3!=a4 and a3!=a1 and a3!=a2):
        cnt=cnt+1
else:
    if(a1!=a3):
        cnt=cnt+1
        if(a2!=a4):
            cnt=cnt+1
    else:
        if(a1!=a4):
            cnt=cnt+1
            if(a2!=a3):
                cnt=cnt+1


print(cnt);
```

```
2 3 4 1
2
```

In [3]:
```
class Date:
    def __init__(self):
        dd=0
        mm=0
        yy=0
    def get(self):
        self.dd=int(input("Enter Day "))
        self.mm=int(input("Enter month "))
        self.yy=int(input("Enter year "))
    def display(self):
        print(self.dd,".",self.mm,".",self.yy)

    def __gt__(self, B):
        if(self.yy>B.yy):
            return True
        else:
            if(self.yy<B.yy):
                return False
            else:
                if(self.mm>B.mm):
                    return True
                else:
                    if(self.mm<B.mm):
                        return False
                    else:
                        if(self.dd>B.dd):
                            return True
                        else:
                            return False

    def __lt__(self, B):
        if(self.yy<B.yy):
            return True
        else:
            if(self.yy>B.yy):
                return False
            else:
                if(self.mm<B.mm):
                    return True
                else:
                    if(self.mm>B.mm):
                        return False
                    else:
                        if(self.dd<B.dd):
                            return True
                        else:
                            return False

    def __eq__(self, B):
        if(self.yy==B.yy and self.mm==B.mm and self.dd==B.dd):
            return True
        else:
            return False



D1=Date()
D1.get()
D2=Date()
D2.get()

print("D1>D2 :", D1 > D2)
print("D1<D2 :", D1 < D2)
print("D1==D2 :", D1 == D2)
```

```
Enter Day 2
Enter month 4
Enter year 1998
Enter Day 2
Enter month 5
Enter year 1998
D1>D2 : False
D1<D2 : True
D1==D2 : False
```

In [6]:
```
class Dist:
    def __init__(self):
        km=m=0
    def get(self):
        self.km=int(input("Enter KM: "))
        self.m=int(input("Enter m: "))
    def display(self):
        print("Distance: ",self.km,":", self.m)
    def __add__(self, B):
        Temp=Dist()
        Temp.m=self.m+B.m
        Temp.km=self.km+B.km
        Temp.km=Temp.km+Temp.m/1000
        Temp.m=Temp.m%1000
        return Temp
    def __sub__(self, B):
        Temp=Dist()
        if self.m<B.m:
            self.m=self.m+1000
            self.km=self.km-1

        Temp.m=self.m-B.m
        Temp.km=self.km-B.km
        return Temp

    def __mul__(self, B):
        Temp=Dist()
        Temp.km = self.km * B.km
        Temp.m = self.m * B.m
        Temp.km += Temp.m // 1000
        Temp.m = Temp.m % 1000
        return Temp

    def __truediv__(self, B):
        Temp=Dist()
        Temp.km = self.km / B.km
        Temp.m = self.m / B.m
        Temp.km += Temp.m // 1000
        Temp.m = Temp.m % 1000
        return Temp
d1=Dist()
d2=Dist()
d3=Dist()

d1.get()
d2.get()
print("D1+D2")
d3=d1 + d2

d3.display()
print("D1-D2")
d3=d1 - d2

d3.display()
d3=d1 * d2
print("D1*D2")
d3.display()
d3=d1 / d2
print("D1/D2")
d3.display()
```

```
Enter KM: 2
Enter m: 200
Enter KM: 3
Enter m: 100
D1+D2
Distance:  5.3 : 300
D1-D2
Distance:  -1 : 100
D1*D2
Distance:  26 : 0
D1/D2
Distance:  0.6666666666666666 : 2.0
```

In [5]:
```
class Box:
    def __init__(self, length, breadth, depth):
        self.length = length
        self.breadth = breadth
        self.depth = depth

    def volume(self):
        return self.length * self.breadth * self.depth

class WeightBox(Box):
    def __init__(self, length, breadth, depth, weight):
        Box.__init__(self, length, breadth, depth)
        self.weight = weight

class ColorWeightBox(WeightBox):
    def __init__(self, length, breadth, depth, weight, color):
        WeightBox.__init__(self, length, breadth, depth, weight)
        self.color = color
box = Box(10, 20, 30)
print("Volume of the box:", box.volume())

weight_box = WeightBox(10, 20, 30, 40)
print("Volume of the weight box:", weight_box.volume())
print("Weight of the weight box:", weight_box.weight)

color_weight_box = ColorWeightBox(10, 20, 30, 40, 'red')
print("Volume of the color weight box:", color_weight_box.volume())
print("Weight of the color weight box:", color_weight_box.weight)
print("Color of the color weight box:", color_weight_box.color)
```

```
Volume of the box: 6000
Volume of the weight box: 6000
Weight of the weight box: 40
Volume of the color weight box: 6000
Weight of the color weight box: 40
Color of the color weight box: red
```

In [ ]: