In [ ]:

```
# Chef with 2 Languages

for i in range(int(input())):
    a,b,a1,b1,a2,b2 = map(int,input().split())
    if (a==a1 or a==b1) and (b==a1 or b==b1):
        print(1)
    elif (a==a2 or a==b2) and (b==a2 or b==b2):
        print(2)
    else:
        print(0)
                                                    #inputs
                                                    # 3
                                                    # 1 2 2 1 3 4
                                                    # 3 4 2 1 4 3
                                                    # 1 2 1 3 2 4
```

```
1
2
0
```

In [ ]:

```
# Create set with defficult level of problem

for t in range(int(input())):
    b=list(map(int,input().split()))
    a=set(b)
    if len(a)==1:
        print(0)
    elif len(a)==2 and b.count(b[0])!=2:
        print(1)
    else:
        print(2)

                                                    # 3
                                                    # 1 4 3 2
                                                    # 4 5 5 5
                                                    # 2 2 2 2
```

```
2
1
0
```

In [ ]:

```
# Develop a python code to check given two dates are equal with overload operaters

from datetime import date
x = date(2013,2,1)
y = date(2013,2,2)

print (x > y)
print (y > x)
```

```
False
True
```

In [ ]:

```
#Operating with two distences (kilomeaters followed by meters)
import math

kilometer1 = int(input("First KM :"))
meter1 = int(input("First MM :"))
kilometer2 = int(input("Second KM :"))
meter2 = int(input("Second MM :"))
print("First distence is",kilometer1,"KM","and",meter1,"Meters\n")
print("Second distence is",kilometer2,"KM","and",meter2,"Meaters\n")

#Addition of both distences
totalkm = (kilometer1+(meter1/1000))+(kilometer2+(meter2/1000))
result = math.modf(totalkm)
dec, integer = result
print("The Addition of the given distence is",integer,"KM","and",dec*1000,"Meters\n")

#substraction of both distences
totalkm1 = (kilometer1+(meter1/1000))-(kilometer2+(meter2/1000))
result1 = math.modf(totalkm1)
dec1, integer1 = result1
print("The Substration of the given distence is",integer1,"KM","and",dec1*1000,"Meters\n")

#multiplacation of both distences
totalkm2 = (kilometer1+(meter1/1000))*(kilometer2+(meter2/1000))
result2 = math.modf(totalkm2)
dec2, integer2 = result2
print("The Multiplication of the given distence is",integer2,"KM","and",dec2*1000,"Meters\n")

#division of both distences
totalkm3 = (kilometer1+(meter1/1000))/(kilometer2+(meter2/1000))
result3 = math.modf(totalkm3)
dec3, integer3 = result3
print("The Division of the given distence is",integer3,"KM","and",dec3*1000,"Meters\n")
```

```
First distence is 10 KM and 222 Meters

Second distence is 11 KM and 333 Meaters

The Addition of the given distence is 21.0 KM and 554.9999999999998 Meters

The Substration of the given distence is -1.0 KM and -111.00000000000065 Meters

The Multiplication of the given distence is 115.0 KM and 845.9259999999915 Meters

The Division of the given distence is 0.0 KM and 901.967704932498 Meters
```

In [ ]:

```
# Creating multi-level inheritance.
class Box:
  def __init__(self,Length,Breadth,Depth):
    self.Length = Length
    self.Breadth = Breadth
    self.Depth = Depth
  def display(self):
    print("Length: ",self.Length)
    print("Breadth: ",self.Breadth)
    print("Depth :",self.Depth)
    volume = (self.Length*self.Breadth*self.Depth)
    print("Volume of the given cube is :",volume)

class WeightBox(Box):
  def __init__(self,Length,Breadth,Depth,Weight):
    Box.__init__(self,Length,Breadth,Depth)
    self.Weight = Weight
  def display(self):
    Box.display(self)
    print("Weight: ",self.Weight)

class Colour(WeightBox):
  def __init__(self,Length,Breadth,Depth,Weight,colour):
    WeightBox.__init__(self,Length,Breadth,Depth,Weight)
    self.colour=colour
  def display(self):
    print("Length: ",self.Length)
    print("Breadth: ",self.Breadth)
    print("Depth: ",self.Depth)
    volume = (self.Length*self.Breadth*self.Depth)
    print("Volume of the given cube is :",volume)
    print("Weight: ",self.Weight)
    print("Colour: ",self.colour)



e = Colour(4,5,6,"2KG","Red")
e.display()
```

```
Length:  4
Breadth:  5
Depth:  6
Volume of the given cube is : 120
Weight:  2KG
Colour:  Red
```