ASSIGNMENT 6

Chef is a software developer, so he has to switch between different languages sometimes. Each programming language has some features, which are represented by integers here. Currently, Chef has to use a language with two given features A and B. He has two options --- switching to a language with two features A1 and B1, or to a language with two features A2 and B2. All four features of these two languages are pairwise distinct. Tell Chef whether he can use the first language, the second language or neither of these languages (if no single language has all the required features)

In [9]:
```
for i in range(eval(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)
```

```
2
1 2 4 9 6 2
8 4 7 6 2 4
```

In [12]:
```
for i in range(int(input())):
    a1, a2, a3, a4 = sorted(map(int, input().split()))
    # sort the difficulties in ascending order to simplify the calculations
    max_sets = 0
    for x in range(a1, a4):
        for y in range(x+1, a4+1):
            if (x == a1 and y == a4) or (x == a2 and y == a3):
                # if the pair includes the highest and lowest difficulties, or the
                continue
            max_sets = max(max_sets, 1)
            for z in range(a1, a4):
                if z in (x, y):
                    continue
                for w in range(z+1, a4+1):
                    if w in (x, y):
                        continue
                    if (z == a1 and w == a4) or (z == a2 and w == a3):
                        # if the second pair includes the highest and lowest diffi
                        continue
                    max_sets = max(max_sets, 2)
    print(max_sets)
```

```
3
1 2 3 4
2
2 2 3 4
1
1 2 2 2
0
```

In [ ]:
```
Develop a python code to check given two dates d1 and d1 , check whether d1 is les
Develop python code to add, subtract , multiply and divide two distances where eac
```

In [15]:
```
class Date:
    def __init__(self, day, month, year):
        self.day = day
        self.month = month
        self.year = year

    def __eq__(self, other):
        if isinstance(other, Date):
            return (self.year, self.month, self.day) == (other.year, other.month,
```

```
        return NotImplemented

    def __lt__(self, other):
        if isinstance(other, Date):
            return (self.year, self.month, self.day) < (other.year, other.month, o
        return NotImplemented

    def __gt__(self, other):
        if isinstance(other, Date):
            return (self.year, self.month, self.day) > (other.year, other.month, o
        return NotImplemented
```

In [16]:
```
d1 = Date(31, 3, 2023)
d2 = Date(1, 4, 2023)

if d1 < d2:
    print("d1 is less than d2")
elif d1 > d2:
    print("d1 is greater than d2")
else:
    print("d1 is equal to d2")
```

```
d1 is less than d2
```

In [17]:
```
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):
        super().__init__(length, breadth, depth)
        self.weight = weight

    def get_weight(self):
        return self.weight

class ColorWeightBox(WeightBox):
    def __init__(self, length, breadth, depth, weight, color):
        super().__init__(length, breadth, depth, weight)
        self.color = color

    def get_color(self):
        return self.color
```

In [18]:
```
# create a Box
box = Box(2, 3, 4)
print("Box volume:", box.volume())

# create a WeightBox
weight_box = WeightBox(2, 3, 4, 5)
print("WeightBox volume:", weight_box.volume())
print("WeightBox weight:", weight_box.get_weight())

# create a ColorWeightBox
color_weight_box = ColorWeightBox(2, 3, 4, 5, "red")
print("ColorWeightBox volume:", color_weight_box.volume())
print("ColorWeightBox weight:", color_weight_box.get_weight())
print("ColorWeightBox color:", color_weight_box.get_color())
```

```
Box volume: 24
WeightBox volume: 24
WeightBox weight: 5
ColorWeightBox volume: 24
ColorWeightBox weight: 5
ColorWeightBox color: red
```

develop python code to add, subtract , multiply and divide two distances where each distance contains two things of the format KM followed by Meters.

In [29]:
```python
def add_distances(dist1, dist2):
    km1, m1 = list(map(int, dist1.split(" KM ")[0])), int(dist1.split(" KM ")[1].s
    km2, m2 = list(map(int, dist2.split(" KM ")[0])), int(dist2.split(" KM ")[1].s
    total_m = (km1[0] * 1000 + m1) + (km2[0] * 1000 + m2)
    total_km, total_m = divmod(total_m, 1000)
    return f"{total_km} KM {total_m} meters"

def subtract_distances(dist1, dist2):
    km1, m1 = list(map(int, dist1.split(" KM ")[0])), int(dist1.split(" KM ")[1].s
    km2, m2 = list(map(int, dist2.split(" KM ")[0])), int(dist2.split(" KM ")[1].s
    total_m = (km1[0] * 1000 + m1) - (km2[0] * 1000 + m2)
    if total_m < 0:
        total_m += 1000
        total_km = (km1[0] - km2[0] - 1)
    else:
        total_km = (km1[0] - km2[0])
    return f"{total_km} KM {total_m} meters"

def multiply_distance(dist, factor):
    km, m = list(map(int, dist.split(" KM ")[0])), int(dist.split(" KM ")[1].split
    total_m = (km[0] * 1000 + m) * factor
    total_km, total_m = divmod(total_m, 1000)
    return f"{total_km} KM {total_m} meters"

def divide_distance(dist, factor):
    km, m = list(map(int, dist.split(" KM ")[0])), int(dist.split(" KM ")[1].split
    total_m = (km[0] * 1000 + m) / factor
    total_km, total_m = divmod(total_m, 1000)
    return f"{total_km} KM {total_m} meters"
```

In [30]:
```python
distance1 = "10 KM 500 meters"
distance2 = "5 KM 250 meters"

# Addition
result = add_distances(distance1, distance2)
print(result)  # Output: 15 KM 750 meters

# Subtraction
result = subtract_distances(distance1, distance2)
print(result)  # Output: 5 KM 250 meters

# Multiplication
result = multiply_distance(distance1, 2)
print(result)  # Output: 21 KM 0 meters

# Division
result = divide_distance(distance1, 2)
print(result)  # Output: 5 KM 250 meters
```

```
6 KM 750 meters
-5 KM -2750 meters
3 KM 0 meters
0.0 KM 750.0 meters
```