

## **Problem #1049**

In the problem, we will be given 3 words, from which we will have to identify the name of a particular animal.

Each element belongs to a particular sequence. The sequence of 3 given words is different for each of the animal.

First of all, we will take inputs for the 3 words.

```
Scanner sc = new Scanner(System.in);
String a = sc.nextLine();
String b = sc.nextLine();
String c = sc.nextLine();
```

Then we can assign all the type names to variables.

```
String type1 = "vertebrado";
String type2 = "ave";
String type3 = "mamifero";
String type4 = "carnivoro";
String type5 = "onivoro";
```

And so on.

After that if we use nested “if” and “else if” to find the particular sequence of 3 words, we will be able to detect the name of the desired animal.

Here is a short snippet of that code.

```
if(a.equals(type1))
{
    if(b.equals(type2))
    {
        if(c.equals(type4))
        {
            System.out.println("aguia");
        }
        else if(c.equals(type5))
        {
            System.out.println("pomba");
        }
    }
    else if(b.equals(type3))
    {
        if(c.equals(type5))
        {
            System.out.println("homem");
        }
        else if(c.equals(type6))
        {
            System.out.println("vaca");
        }
    }
}
```

## Problem #1051

In this problem we have to find the amount of tax a person has to pay depending on his/her salary. In the question, a table is given which shows the percentage of tax one has to pay according to his salary range.

For example, say a person earns 4,800 R\$. For the first 2000 R\$, the person won't have to pay any tax on. Then for the next 1000 R\$, the person will have to pay 8% of 1000. Then for the next 1500 R\$, the person will have to pay 18% of 1500. And then for the last 300 R\$, the person will have to pay 28% of 300. The total sum of all these amounts, is the amount the person will have to pay as tax due to his salary.

First we will take the salary as input. If the salary is not more than 2000, we will simply print "Isento".

```
if (x <= 2000) {  
    System.out.println("Isento");  
}
```

Then we will subtract 2000 as there is no tax on it. Then if the value of the variable is still greater than 0, we will check if it is less than 1000 or not. If it is less than 1000, we will impose 8% tax on the remaining amount and if it is not less or equal to 1000, we will impose 8% tax on 1000.

```
remainder = x - 2000;  
if (remainder > 0) {  
    if (remainder <= 1000) {  
        tax1 = remainder * 0.08;  
        System.out.printf("R$ %.2f%n", tax1);  
    } else {  
        tax1 = 1000 * 0.08;  
    }  
    remainder = remainder - 1000;  
}
```

The exact same process will go for the next ranges of salary according to the table given.

```
if (remainder > 0) {  
    if (remainder <= 1500) {  
        tax2 = remainder * 0.18;  
        totalTax = tax1 + tax2;  
        System.out.printf("R$ %.2f%n", totalTax);  
    } else {  
        tax2 = 1500 * 0.18;  
    }  
    remainder = remainder - 1500;  
}  
if (remainder > 0) {  
    tax3 = remainder * 0.28;  
    totalTax = tax1 + tax2 + tax3;  
    System.out.printf("R$ %.2f%n", totalTax);  
}
```

## **Problem #1096**

In this problem, we have to print a pattern of numbers. Basically, we will have to print 2 numbers, I and J.

The value of I will be all the odd numbers from 1 to 9 inclusive. But for each value of I, there will be 3 output lines. The 3 output lines will consist of the present value of I but the value of J will vary on those 3 lines. The value of J will be 7, 6, and 5 on the first, second and third line respectively.

To solve this problem, we will use two nested for loops. The first for loop will traverse all the odd numbers from 1 to 9 inclusive and will contain the value of I. And the inner for loop will iterate thrice every time. Inside the inner for loop, we will print those 3 lines. For the first line, we will print the current value of I and '7' as the value of J. Similarly, for the second and third lines, we will print the current value of I and '6' and '5' as the value of J respectively.

The code will look like this.

```
for i in range (1, 10, 2):  
    for j in range(3):  
        if j == 0: print(f'I={i} J=7')  
        elif j == 1: print(f'I={i} J=6')  
        else: print(f'I={i} J=5')
```

## Problem #1097

In this problem, we have to print a pattern of numbers. Basically, we will have to print 2 numbers, I and J.

The value of I will be all the odd numbers from 1 to 9 inclusive. But for each value of I, there will be 3 output lines. The 3 output lines will consist of the present value of I but the value of J will vary on those 3 lines. Initially, the value of J will be 7. For the first value of I, the value of J will decrement twice. For the second value of I, the value of J will increment by 2 from the initial value of J. So for the second value of I, the value of J will now be  $7 + 2 = 9$ . And then again, the value of J will decrement twice as there are 3 outputs for each value of I. But then again, for the next value of I, the value of J will again increment by 2 from the previous initial value which was 9. So now the value of J for the 3<sup>rd</sup> value of I will be  $9 + 2 = 11$ . This process will keep continuing for all the values of I.

So first we will initialize the value of J to 7. Then we will use two nested loops. The outer loop will iterate through all the values of I. The inner loop will iterate thrice as for each value of I, we will have to print 3 lines of outputs. Inside the inner loop, we will print the current values of I and J and decrement the value of J by 1. Because, for a value of I, J will decrease twice. After the inner loop is finished, to restore the value of J to previous initial value and also to add 2 to it for the next value of I, what we will do is add 5 to the value of current J i.e.  $j = j + 5$ . This will restore the decremented value of j and will also increment the value of j by 2 from the initial value.

So the code will look like this.

```
j = 7
for i in range(1, 10, 2):
    for c in range(3):
        print(f'I={i} J={j}')
        j -= 1
    j += 5
```

## Problem #1098

In this problem the question is a little bit different than the previous ones. Here the value of I will range from 0 to 2 inclusive, but the increment of I will be 0.2. That is the values of I will be 0, 0.2, 0.4, ..... , 1.6, 1.8, 2.0.

And for each value of I, there will be 3 values of J. The initial 3 values of J are 1, 2 and 3. For each values of I, these 3 values will also increment by 0.2. For example, when I = 0.2, the 3 values of J will be 1.2, 2.2, 3.2 respectively.

One more thing about the output is that, when the values of I and J are whole numbers, the output will be as whole number. But when the values of I and J are not integers, the output will show up to 1 decimal places.

In python, iteration of decimal values cannot be done in a for loop. So I had to use a while loop. First, I initialized the value of I, and also the initialized the 3 different values of J in 3 different variables.

```
i = 0.0
x, y, z = 1.0, 2.0, 3.0
```

Then until the values of I doesn't cross 2, I kept iterating and printing the values of I and J. I also kept incrementing the values of i, x, y, z by 0.2.

But the catch in the problem was printing in the correct format. To do so, I used a condition to separate the whole numbers. If the current value of I is a whole number, I had to print in one way and if it was not whole number, I had to print the values in another way.

The code with which I did so is attached below.

```
while i <= 2:
    if i.is_integer():
        print(f'I={int(i)} J={int(x)}')
        print(f'I={int(i)} J={int(y)}')
        print(f'I={int(i)} J={int(z)}')
    else:
        i = round(i, 1)
        x = round(x, 1)
        y = round(y, 1)
        z = round(z, 1)
        print(f'I={i} J={x}')
        print(f'I={i} J={y}')
        print(f'I={i} J={z}')
    x += 0.2; y += 0.2; z += 0.2
    i += 0.2
```