

1. Discount Calculator

The iron and steel industry manufactures various products using iron, steel, tungsten, and nickel. Customers are the one who place bulk orders to the industry. The industry gives two discount cards for its customers. One is based on the type of metal purchased and another one is based on the total purchase cost. At any point in time, customer can use one of the two discount cards.

The goal is to calculate the discount using the two discount cards and recommend the customer to use the one which gives the highest discount.

For example, the industry has the below two discount cards

Type of metal purchased	Discount% on purchase amount
Iron	7%
Steel	3%
Tungsten	2%
Nickel	1%

Total purchase cost in Rs	Discount% on the purchase cost
Till 25000	Nil
25001 to 50000	5%
50001 to 100000	7%
>100000	10%

If the customer purchases Rs.15000 worth of Iron, Rs.10000 worth of steel, Rs.2000 worth of Tungsten and Rs.1500 worth of Nickel, then as per the two discounts rate cards, then the discount can be calculated as follows,

As per discount rate card 1, discount rate = $(15000 * 7 / 100) + (10000 * 3 / 100) + (2000 * 2 / 100) + (1500 * 1 / 100) = \text{Rs.}1405.00$

As per discount rate card 2, total purchase amount = $15000+10000+2000+1500=28500$, hence discount rate = $28500*5/100 = \text{Rs.}1425.00$

As the discount is more using the second discount rate, the customer can use the second discount rate card and second discount amount.

Function Description

Complete the function calculateDiscount in the editor and it prints the discount rate.

calculateDiscount has the following parameter(s);

Purchase cost of Iron: integer

Purchase cost of Steel: integer

Purchase cost of Tungsten: integer

Purchase cost of Nickel: integer

Constraints

- All the four input values are integers
- Final output should be printed in a float with two precisions

Input Format For Custom Testing

The first line contains an integer, n_1 , denoting the purchase cost of iron

The second line contains an integer, n_2 , denoting the purchase cost of steel

The third line contains an integer, n_3 , denoting the purchase cost of tungsten

The fourth line contains an integer, n_4 , denoting the purchase cost of nickel

Sample Input

15000

10000

2000

1500

Sample Output

1425.00

Explanation

As per discount rate card 1, discount rate = $(15000 \times 7/100) + (10000 \times 3/100) + (2000 \times 2/100) + (1500 \times 1/100) = \text{Rs.}1405.00$

As per discount rate card 2, total purchase amount = $15000 + 10000 + 2000 + 1500 = 28500$, hence discount rate = $28500 \times 5/100 = \text{Rs.}1425.00$

Higher discount is Rs.1425.00

Solution:

```
import java.util.Scanner;

public class DiscountCalculator {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int iron = scan.nextInt();
        int steel = scan.nextInt();
        int tun = scan.nextInt();
        int nic = scan.nextInt();

        calculateDiscount(iron, steel, tun, nic);

    }

    static void calculateDiscount(int i, int s, int t, int n) {
```

```
float d1 = 0, d2 = 0;

d1 = i * (7.0f / 100) + s * (3.0f / 100) + t * (2.0f / 100) + n * (1.0f / 100);
float total = i + s + t + n;

if (total >= 25001 && total <= 50000) {
    d2 = total * (5.0f / 100);
} else if (total >= 50001 && total <= 100000) {
    d2 = total * (7.0f / 100);
} else if (total > 100000) {
    d2 = total * (10.0f / 100);
}

if (d1 > d2) {
    System.out.println(d1);
} else {
    System.out.println(d2);
}

}

}
```

2. Triangle Game

The Westland Game Fair is the premier event of its kind for kids interested in some intellectual and cognitive brain games. Exciting games were organized for kids between age group of 8 and 10. One such game was called the "Triangle game", where different number boards in the range 1 to 180 are available. Each kid needs to select three number boards, where the numbers on the boards correspond to the angles of a triangle.

If the angles selected by a kid forms a triangle, he/she would receive Prize 1. If the angles selected by a kid forms a right triangle, he/she would receive Prize 2. If the angles selected by the kids form an equilateral triangle, he/she would receive Prize 3. If the angles selected by a kid do not form even a triangle, then he/she will not receive any prizes. Write a program for the organizers to fetch the result based on the number boards selected by the kids.

Input Format:

There are 3 lines in the input, each of which corresponds to the numbers on the boards that the kids select.

Output Format:

The output should display "Prize 1" or "Prize 2" or "Prize 3" or "No Prize" based on the conditions given.

Refer sample input and output for formatting specifications.

Sample Input 1:

60

50

70

Sample Output 1:

Prize 1

Sample Input 2:

60

60

70

Sample Output 2:

No Prize

Solution:

```
import java.util.Scanner;

public class TriangleGame {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();

        if (a + b + c == 180) {
            if (a == 60 && b == 60 && c == 60) {
                System.out.println("Prize 3");
            } else if (a == 90 || b == 90 || c == 90) {
                System.out.println("Prize 2");
            } else {
```

```
        System.out.println("Prize 1");
    }
} else {
    System.out.println("No Prize");
}

}

}
```

3. Count Divisors

You have been given 3 integers - l , r , and k . Find how many numbers between l and r (both inclusive) are divisible by k . You do not need to print these numbers, you just have to find their count.

Input Format

The first and only line of input contains 3 space-separated integers l , r , and k .

Output Format

Print the required answer on a single line.

Sample Input:

1 10 1

Sample Output:

10

Solution:

```
import java.util.Scanner;

public class CountDivisors {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int l = scan.nextInt();
        int r = scan.nextInt();
        int k = scan.nextInt();

        int count = 0;
        for (int i = l; i <= r; i++) {
            if (i % k == 0) {
                count++;
            }
        }

        System.out.println(count);
    }
}
```


}

}