

1. Problem Description:

Sajid was booking a train ticket from Chennai to Delhi for his family.

Two of the relatives was interested in joining that journey from different places with their family members

So, Sajid booked tickets for those persons also along with his family members.

He wants to know the total number of tickets for this travel. Can you help him in finding the total number of passengers?

Constraint:

$$1 < \text{num1} \leq 15$$

$$1 \leq \text{num2} \leq 15$$

$$1 \leq \text{num3} \leq 15$$

Input Format:

Only Line of input has three integers num 1, num2 and num3 separated by a space representing the numbers of ticket booked by Sajid at three different interval of time.

Output Format:

Print the total number of tickets booked by Sajid.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int a = input.nextInt();

        int b = input.nextInt();

        int c = input.nextInt();

        int d = a+b+c;

        System.out.println(+d);

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

5 10 15

EXPECTED OUTPUT

30

Test Case 2

INPUT (STDIN)

3 5 6

EXPECTED OUTPUT

14

2. Problem Description:

Nancy bought apples in a fruit shop.

The shop keeper specified the the bill amount. Nancy also given some amount to the shop keeper for paying the bill.

But she likes to know the quotient and remainder after dividing the amount given by her by the bill amount specified by shop keeper.

Can you help nancy in finding it?

Constraint:

5 samtgivens 2500

$5 < \text{billamt} < 2500$

Input Format:

First Line: Integer value of amtgiven representing the amount given by nancy.

Second Line: Integer value of

billamt representing the amount specified by
the shop keeper

Output Format:

First Line: Print the Quotient in integer format.

Second Line: Print the Remainder in integer format.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int amtgiven = input.nextInt();

        int billamt = input.nextInt();

        int q = amtgiven/billamt;

        int r = amtgiven%billamt;

        System.out.println("Quotient:"+q);

        System.out.println("Remainder:"+r);

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
600
520
```

EXPECTED OUTPUT

```
Quotient:1
Remainder:80
```

Test Case 2

INPUT (STDIN)

```
789
256
```

EXPECTED OUTPUT

```
Quotient:3
Remainder:21
```

3. Problem Description:

Rathik organized technical round interview in Macrosoft for the set of computer science candidates.

The problem is to perform addition, subtraction, multiplication, and division of given two numbers.

Rathik have given the deadline of only 5 minutes to complete the problem.

Can you Help the candidates to complete the problem within the specified time limit?

Constraint:

$1 < \text{testnum1} < 50$

$1 \leq \text{testnum2} < 50$

Input Format:

The only line of input has two numbers a and b of type integers separated by a comma.

Output Format:

Print Addition, Subtraction, Multiplication, Division, and Modulus of given two numbers in a separate line respectively.

Note: Rathik instructed his candidates to print the result of the division with 3 values after decimal point.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int testnum1 = input.nextInt();
        int testnum2 = input.nextInt();

        System.out.println("Addition : "+ (testnum1+testnum2));
        System.out.println("Subtraction : "+(testnum1-testnum2));
        System.out.println("Multiplication : "+(testnum1*testnum2));
        float div;
        div = (float) testnum1 / testnum2;
        System.out.println("Division : "+String.format("%.3f",div));
        System.out.println("Modulus : "+(testnum1%testnum2));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
17 13
```

EXPECTED OUTPUT

```
Addition : 30  
Subtraction : 4  
Multiplication : 221  
Division : 1.308  
Modulus : 4
```

Test Case 2

INPUT (STDIN)

```
31 4
```

EXPECTED OUTPUT

```
Addition : 35  
Subtraction : 27  
Multiplication : 124  
Division : 7.750  
Modulus : 3
```

4. Problem Description:

Rebecca bought a new volleyball in the sports shop. It looks like a medium size.

She somehow found the radius of the sphere. But she would like to know the volume of that ball. Can you help him in finding the Volume of the ball?

Functional Description:

Volume = $(4.0/3.0) \times \pi \times r^3$

$\pi = 3.14$

Constraint:

$1.00 \leq r \leq 5.00$

Input Format:

The only line of input has a single value of type float representing the radius of the ball.

Output Format:

Print the volume of the ball in a single line.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        float radiusofball = input.nextFloat();

        double v = (4.0/3.0)*3.14 ;

        System.out.println(String.format("%.6f",v*(Math.pow(radiusofball, 3))));

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

2.56

EXPECTED OUTPUT

70.240606

Test Case 2

INPUT (STDIN)

3.1

EXPECTED OUTPUT

124.724975

5. Problem Description:

Selvan was playing with the a object of random size for stress relief.

Selvan knows that the Length, Width, and Height of the object.

But he would like to know the surface area of the object he is playing with.

Can you help him in finding it?

Functional Description:

Surface area of the Object = $2 \times [\text{width} \times \text{length} + \text{length} \times \text{height} + \text{height} \times \text{width}]$

Constraints:

$1 \leq \text{width} \leq 10$

$1 \leq \text{height} \leq 10$

Input Format:

First Line: Length of the object in Integer.

Second Line : Width of the object in Integer

Third Line: Height of the object in Integer

Output Format:

Print a single integer value representing the surface area of the object selvam is playing with.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int length = input.nextInt();

        int width = input.nextInt();

        int height = input.nextInt();

        int area = 2*(width*length + length*height + height*width);

        System.out.println(area);

    }

}
```


✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

3
2
3

EXPECTED OUTPUT

42

Test Case 2

INPUT (STDIN)

3
7
4

EXPECTED OUTPUT

122

6. Problem Description:

The Electricity Officer has mentioned the total counts of unit and amount.

The officer inform the customer the bill amount in a unique format.

The format given by electricity officer as follow:

But customers are finding the difficult to find the exact amount that needs to be paid.

Can you help the customers?

Functional Description:

Total Bill Amount = $\text{unitconsumed} \wedge \text{costperunit}$

Constraints:

$1 \leq \text{unitconsumed} \leq 500$

$2 \leq \text{costperunit} \leq 10$

Input Format:

The first line of input represents the integer value of unitconsumed

The second line of input represents the integer value of costperunit

Output Format:

Print the total Bill amount in single line.

Code:

```
import java.io.*;
import java.lang.Math;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int unitconsumed = input.nextInt();
        int costperunit = input.nextInt();
        System.out.println(Math.pow(unitconsumed, costperunit));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

23
3

EXPECTED OUTPUT

12167.0

Test Case 2

INPUT (STDIN)

19
5

EXPECTED OUTPUT

2476099.0

7. Problem Description:

Phoenix mall in the capital city of Washington and it is rectangular in shape when it is seen on the map with the size $n \times m$ meters.

On the occasion of the jubilee anniversary, a decision was taken to pave the Square with square marbles stones. Each stone is of the size $n \times n$.

Can you what is the least number of stones needed to pave the Square?

It's allowed to cover the surface larger than the Mall Square, but the Square has to be covered.

It's not allowed to break the stones. The sides of stones should be side by side(parallel) to the sides of the Square.

Constraints:

$$1 \leq n \leq 10^9$$

$$1 \leq m \leq 10^9$$

$$1 \leq a \leq 10^9$$

Input Format:

The only line of input contains three positive integer numbers n , m and a separated by a space.

Output Format:

Print the needed number of stones.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        int m = input.nextInt();
        int a = input.nextInt();
        System.out.println(((m+a-1)/a)*((n+a-1)/a));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

6 6 4

EXPECTED OUTPUT

4

Test Case 2

INPUT (STDIN)

8 4 4

EXPECTED OUTPUT

2

8. Problem Description:

During the IPL Match between CSK and MI, as a part of IPL contest the question was asked to the fans.

Who are all giving the correct answer to that question will get the free VIP box ticket for the Final for which CSK have already qualified.

The question is convert given integer number to octal and hexadecimal number respectively.

Abilash is an die heart CSK fan. Can you help him answer the question so that he can watch CSK play the final from VIP box?

Constraints:

1 ≤ ipIno ≤ 10000

Input Format:

Only line of input has single integer number that need to be converted.

Output Format:

In the First line of output print the octal number equivalent to the input value.

In the Second line of output print the hexadecimal number equivalent to the input value.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int ipIno = input.nextInt();
        System.out.println(Integer.toOctalString(ipIno));
        System.out.println(Integer.toHexString(ipIno));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1953

EXPECTED OUTPUT

3641
7a1

Test Case 2

INPUT (STDIN)

8751

EXPECTED OUTPUT

21057
222f

9. Problem Description:

On one beautiful Sunday Selvan went to Aaron's house for exam preparation.

They have decided to study Mathematics subject because they have exams by coming Monday, Aaron is a master in Mathematics but Selvan is not so good in Mathematics so James trained with Selvan for getting a high score in the exam.

After teaching some problems to Selvan. Aaron have given some tasks to Selvan to solve.

The problem is to convert input float into a double.

Can you help Selvan in finding the solution?

Constraints:

$1.00 \leq \text{num1} \leq 100.00$

$1.00 < \text{num2} \leq 100.00$

$1.00 \leq \text{resnum 1} \leq 100.00$

$1.00 \leq \text{resnum2} < 100.00$

Input Format:

The first and second line of the input represents two different input value of type float. Output Format:

The first and second line of the output represents outputs of first and second line of input of type double.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        float num1 = input.nextFloat();

        float num2 = input.nextFloat();

        System.out.print(num1);

        System.out.println("0000");

        System.out.print(num2);

        System.out.println("0000");

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

3.23

5.67

EXPECTED OUTPUT

3.230000

5.670000

Test Case 2

INPUT (STDIN)

48.18

17.77

EXPECTED OUTPUT

48.180000

17.770000

10. Problem Description:

The employees of one million dollar profit company TeamZilla organised the strike because they want to have additional salary increment, the strike is continuing for more than a month now.

Rathik the CEO of TeamZilla has found the solution to break the strike, so he organised a small technical competition for his employees.

Most of the employees who were part of the strike have participated in the technical event announced and in that there was a task of printing the ASCII Value of the character inputted.

Can you help them to complete the task and win the competition?

Constraint:

$a \leq \text{Asc} \leq z$

$A \leq \text{Asc} \leq Z$

Input format:

Only Line of input represents a single alphabetic character.

Output format:

Print the integer ASCII value corresponding to the input alphabet.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        char Asc = input.next().charAt(0);
        System.out.println(+Asc);
    }
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
g	Y
EXPECTED OUTPUT	EXPECTED OUTPUT
103	89

11. Question description:

Kenneth is playing in a T20 cricket match. In a match, Team A plays for 20 overs. In a single over, the team gets to play 6 times, and in each of these 6 tries, they can score a maximum of 6 runs. After Team A's 20 overs are finished, Team B similarly plays for 20 overs and tries to get a higher total score than the first team. The team with the higher total score at the end wins the match.

Kenneth is in Team B. Team A has already played their 20 overs, and have gotten a score of R. Kenneth's Team B has started playing, and have already scored C runs in the first O overs. In the remaining 20-O overs, find whether it is possible for Kenneth's Team B to get a score high enough to win the game. That is, can their final score be strictly larger than R?

Constraints:

- $0 \leq C \leq R \leq 720$
- $0 \leq C \leq 36 * O$

Input Format:

There is a single line of input, with three integers, R,O,C

Output Format:

Output in a single line, the answer, which should be "YES" if it's possible for Kenneth's Team B to win the match and "NO" if not.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main (String[] args)
    {
        int r,o,c;

        Scanner sc = new Scanner(System.in);

        r = sc.nextInt();
        o = sc.nextInt();
        c = sc.nextInt();

        int a = 20-o;

        int b = (a*6*6)+c;

        String d = (r<b) ? "YES" : "NO";

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

```
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

719 18 648

EXPECTED OUTPUT

YES

Test Case 2

INPUT (STDIN)

4000 10 40

EXPECTED OUTPUT

NO

12. Question description

Zach and Miller are preparing themselves for programming contests.

An important part of preparing for a competition is sharing programming knowledge from the experienced members to those who are just beginning to deal with the contests. Therefore, during the next team training Zach decided to make teams so that newbies are solving problems together with experienced participants.

Zach believes that the optimal team of three people should consist of one experienced participant and two newbies. Thus, each experienced participant can share the experience with a large number of people.

However, Miller believes that the optimal team should have two experienced members plus one newbie. Thus, each newbie can gain more knowledge and experience.

As a result, Zach and Miller have decided that all the teams during the training session should belong to one of the two types described above. Furthermore, they agree that the total number of teams should be as much as possible.

There are n experienced members and m newbies on the training session. Can calculate what maximum number of teams can be formed? you

Constraints:

$0 \leq n, m \leq 5 \cdot 10^5$

Input Format:

The first line contains two integers n and m - the number of experienced participants and newbies that are present at the training session.

Output Format:

Print the maximum number of teams that can be formed.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args)

    {

        Scanner input = new Scanner(System.in);

        int n,m;

        n = input.nextInt();

        m = input.nextInt();

        int c = (n+m)/3;

        System.out.println(Math.min(Math.min(n,m),Math.min(m,c)));
```

```
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

4 5

EXPECTED OUTPUT

3

Test Case 2

INPUT (STDIN)

1231 1253

EXPECTED OUTPUT

828

13. Question description

Once upon a time, there lived a mad programmer. He loved to solve creative problems other than anything. His wife loved him quite a lot but disliked his curiosity for the problems. One day he came from office, his wife gave him a glass of cold lime juice. She was in a romantic mood and waiting for some romantic stuff. But the programmer asked her curiously, "If I give u radius of the top and bottom part of the glass and the height, can you come up with the volume of the glass?" His wife became a bit disappointed but as she is smart she replied with a smile, "You already have drunk some juice, and the glass is not full. If I give you the height of the juice, can you find the volume of the remaining juice in the glass?" Then the programmer kissed his wife and said, "You are the best problem setter in the world!"

Now he set the same problem for you. The radius of the upper part r_1 and lower part r_2 is given. If height of the glass is h and height of the juice is p what is the volume of the juice in the glass?

problem-1217-juice-1603773420519.png (153x172)

Constraints:

$$1 \leq r_1 < r_2 \leq 100$$

$$1 \leq p \leq h \leq 100$$

Input Format:

Each case starts with a line containing four integers r_1, r_2, h and p

Output Format:

For each case, print the case number and the volume of the juice in the glass, and the decimal value should be rounded off to 4 decimal places

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double r1,r2,h,p,r,k,juice;

        r1=sc.nextDouble();

        r2=sc.nextDouble();

        h=sc.nextDouble();

        p=sc.nextDouble();

        r = p/(h*1.0)*(r1-r2)+r2;

        k = (Math.PI/3.0);

        juice = k*p*(r*r+r2*r2+(r*r2));
```

```
        System.out.printf("%.4f\n",juice);
    }
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
5 2 3 3	9 8 10 12
EXPECTED OUTPUT	EXPECTED OUTPUT
122.5221	2792.7502

14. Question description

Bryan Mills needs a fence around his farm, but he is too lazy to build it himself. So he purchased a fence-building robot.

He wants the fence to be a regular polygon. The robot builds the fence along a single path, but it can only make fence corners at a single angle a .

Will the robot be able to build the fence Bryan Mills wants? In other words, is there a regular polygon which angles are equal to a ?

Constraints:

$0 < a < 180$

Input Format:

Single line input contains a single integer a - the angle the robot can make corners at measured in degrees.

Output Format:

Output on a single line "YES" (without quotes), if the robot can build a fence Bryan Mills wants, and "NO" (without quotes), if it is impossible.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int y,x;
        x = input.nextInt();
        y = (360%(180-x));
        System.out.println(y==0 ? "YES": "NO");
    } }
```

✓ Logical Test Cases

Test Case 1
INPUT (STDIN)
30
EXPECTED OUTPUT
NO

Test Case 2
INPUT (STDIN)
60
EXPECTED OUTPUT
YES

15. Question description

Charles has three socks in his drawer. Each sock has one of 10 possible colors, which are represented by integers between 1 and 10. Specifically, the colors of the socks are A,B and C

Charles has to wear two socks which have the same color. Help Charles find out if that is possible or not. Charles has three socks in his drawer. Each sock has one of 10 possible colors, which are represented by integers between 1 and 10. Specifically, the colors of the socks are A,B and C

Charles has to wear two socks which have the same color. Help Charles find out if that is possible or not.

Explanation:

Since there are no two socks with the same color, Charles cannot wear a pair of socks with the same color.

Constraints

$1 < A, B, C < 10$

Input Format:

The first and only line of the input contains three space-separated integers A,B and C

Output Format:

Print a single line containing the string "YES" if it is possible for Charles to wear two socks with the same color or "NO" if it is impossible.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main (String[] args){
        int a,b,c;

        Scanner sc = new Scanner(System.in);

        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();

        if(a==b || b==c || c==a)
        {
            System.out.println("YES");
        }
        else
```

```
{  
    System.out.println("NO");  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

5 5 5

EXPECTED OUTPUT

YES

Test Case 2

INPUT (STDIN)

4 5 8

EXPECTED OUTPUT

NO

16. Question description

John has a little brother and his name is Mathew.

Mathew is really young and is just starting to learn english, John wants to explain him about consonants and vowels

Help john to write a program that takes a letter as input and displays whether the entered character is a consonant or vowel

Constraints

$1 \leq N \leq 10^5$

$|S| = N$

Input Format:

Single line input has a character input of single letter

Output Format:

Print whether it is a "Vowel" or "Consonant"

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner (System.in);

        char ch = input.next().charAt(0);

        if(ch=='A' || ch=='a' || ch=='E' || ch=='e' || ch=='I' || ch=='i' || ch=='O' || ch=='o') {

            System.out.println("Vowel");

        }

        else{

            System.out.println("Consonant");

        }

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

A

EXPECTED OUTPUT

Vowel

Test Case 2

INPUT (STDIN)

e

EXPECTED OUTPUT

Vowel

17. Question description

Jones has got a task from his boss and he needs to finish it by evening

He gave him a set of phone numbers and he has to reverse the digits of those phone numbers and generate new phone numbers which are going to be used by the company

Help Jones to complete the task quick since he wants to watch the football match this evening

Constraints

$1 \leq n \leq 100$

$1 \leq k \leq 10$

Input Format:

First line, contains n number of test cases

Followed by no. of phone numbers

Output Format:

Reversed digits of n phone numbers in separate lines

Code:

```
import java.util.Scanner;

public class Class815427160771
{
    int n,remainder,reverse;

    static int reverse(int n){
        int reverse = 0;
        while(n>0){
            int remainder = n%10;
            reverse = (reverse*10) + remainder;
            n /=10;
        }
        return reverse;
    }

    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while(t-->0){
```

```

        int n = sc.nextInt();
        System.out.println(reverse(n));
    }
}

```

Logical Test Cases

Test Case 1

INPUT (STDIN)

```

4
27234847
67875757
57585884
45787844

```

EXPECTED OUTPUT

```

74843272
75757876
48858575
44878754

```

Test Case 2

INPUT (STDIN)

```

6
27234238
83784378
37247347
37437738
84754843
73403738

```

EXPECTED OUTPUT

```

83243272
87348738
74374273
83773473
34845748
83730437

```

18. Question description

Tim went camping with his friends and they thought to casually roam around in the nature

They come across a lake and they thought to play around for a while so they play a small game

There are three piles of stones. The first pile contains a stones, the second pile contains

b stones and the third pile contains c stones. Tim must choose one of the piles and split

the stones from it to the other two piles, specifically, if the chosen pile initially contained

s stones, he should choose an integer k ($0 \leq k \leq s$), move k stones from the chosen pile

onto one of the remaining two piles and $s - k$ stones onto the other remaining pile.

Determine if it is possible for the two remaining piles (in any order) to contain x stones

and y stones respectively after performing this action

Help Tim find the answer to this question.

Constraints:

$$1 \leq T \leq 100$$

$$1 \leq a, b, c, x, y \leq 10^6$$

Input Format:

The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.

The first and only line of each test case contains five space-separated integers a , b , c , x and y

Output Format:

For each test case, print a single line containing the string "YES" if it is possible to obtain piles of the given sizes or "NO" if it is impossible.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int t = input.nextInt();

        for(int i=0;i<t;i++)

        {

            int a = input.nextInt();

            int b = input.nextInt();

            int c = input.nextInt();
```

```

        int x = input.nextInt();

        int y = input.nextInt();

        if((a+b+c!=(x+y) || Math.min(x,y)<Math.min(a,Math.min(b,c)))

        System.out.println("NO");

        else

        System.out.println("YES");

    }

}
}

```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

4

1 2 3 2 4

3 2 5 6 5

2 4 2 6 2

6 5 2 12 1

EXPECTED OUTPUT

YES

NO

YES

NO

Test Case 2

INPUT (STDIN)

2

7 8 9 2 3

8 7 7 9 9

EXPECTED OUTPUT

NO

NO

19. Question description:

Stephen has three shoes of the same size lying around. Each shoe is either a left shoe (represented using 0) or a right shoe (represented using 1). Given A, B, C, representing the information for each shoe, find out whether he can go out now, wearing one left shoe and one right shoe.

Constraints:

$1 \leq T \leq 8$

$0 \leq A, B, C \leq 1$

Input Format:

The first line contains an integer T, the number of test cases. Then the test cases follow

Each test case contains a single line of input, three integers A, B, C.

Output Format:

For each test case, output in a single line the answer: 1 if it's possible to go out with a pair of shoes and 0 if not.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main (String[] args)
    {
        Scanner input = new Scanner(System.in);
        int t = input.nextInt();
        while(t-->0)
        {
            int a,b,c;
            a = input.nextInt();
            b = input.nextInt();
            c = input.nextInt();
            int n = a+b+c;
            if(n>=1&& n<3)
                System.out.println("1");
            else
                System.out.println("0");
        }
    }
}
```



```
    }  
  }  
}
```

▼ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3  
0 0 0  
0 1 1  
1 0 1
```

EXPECTED OUTPUT

```
0  
1  
1
```

Test Case 2

INPUT (STDIN)

```
4  
1 0 1  
1 1 0  
0 0 0  
1 0 1
```

EXPECTED OUTPUT

```
1  
1  
0  
1
```

20. Question description

You are given the height H (in metres) and mass M (in kilograms) of Tony. The

Body Mass Index (BMI) of a person is computed as $M/(H^2)$

Report the category into which Tony falls, based on his BMI:

Category 1: Underweight if BMI < 18

Category 2: Normal weight if BMI is $\{19, 20, \dots, 24\}$ • Category 3: Overweight if BMI is $\{25, 26, \dots, 29\}$

- Category 4: Obesity if BMI > 30

Constraints:

$$1 < T < 2 \cdot 10^4$$

$$1 < M \leq 10^4$$

$$1 < H < 10^2$$

Its guaranteed that H^2 divides M

Input Format:

- The first line of input will contain an integer, T , which denotes the number of testcases. Then the testcases follow.
- Each testcase contains a single line of input, with two space separated integers, M , H , which denote the mass and height of Tony respectively.

Output Format:

For each testcase, output in a single line, 1, 2, 3 or 4, based on the category in which Tony falls.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int t = input.nextInt();
        while(t-->0)
        {
            float bmi,m,h;
            m = input.nextFloat();
            h = input.nextFloat();
            bmi = m/(h*h);
```

```

        if(bmi<=18)
        {System.out.println("1");}
        else if(bmi>18 && bmi<=24)
        {System.out.println("2");}
        else if(bmi>24 && bmi<=29)
        {System.out.println("3");}
        else
        {System.out.println("4");}
    }
}

```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```

3
72 2
80 2
120 2

```

EXPECTED OUTPUT

```

1
2
4

```

Test Case 2

INPUT (STDIN)

```

4
80 1.8
76 1.6
50 1.2
90 1.8

```

EXPECTED OUTPUT

```

3
4
4
3

```

21. Question description

Bruce has just entered a lottery contest where he gets a chance to win 1000\$

He must answer a simple question to qualify for the next round

He has to count the number of multiples of 7 between 1 and the given number n

Can you help him to win this lottery so that he can have a nice summer vacation?

Constraints

$1 \leq n \leq 10^5$

Input Format:

Single line input has integer value n

Output Format:

Print the number of multiples of 7 between 1 to n

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        long n,res = 0;

        n = input.nextLong();

        for (long i = 1; i <= n; i++)

            if (i % 7 == 0)

                res++;

        System.out.println(res);

    }

}
```

✓ Logical Test Cases

Test Case 1
INPUT (STDIN)
38378488
EXPECTED OUTPUT
5482641

Test Case 2
INPUT (STDIN)
45794743
EXPECTED OUTPUT
6542106

22. Question description

Bill was solving some problems in his math textbook and after he was done his friend

Ray gave him a challenge

He gave him the sizes of angles of a simple quadrilateral (in degrees) A,B,C and D in some order along its perimeter. Bill has to determine whether the quadrilateral is cyclic.

Note: A quadrilateral is cyclic if and only if the sum of opposite angles is 180 degrees

Constraints:

$$1 \leq T \leq 10^4$$

$$1 < A, B, C, D \leq 357$$

$$A + B + C + D = 360$$

Input Format:

The first line of the input contains a single integer T denoting the number of test cases.

The description of T test cases follows.

The first and only line of each test case contains four space-separated integers A, B, C and D.

Output Format:

Print a single line containing the string "YES" if the given quadrilateral is cyclic or "NO" if it is not (without quotes).

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int A,B,C,D;
        int t = input.nextInt();
        while(t-->0)
        {
            A = input.nextInt();
            B = input.nextInt();
            C = input.nextInt();
            D = input.nextInt();
            if(A+C==180 && B+D==180)
```

```

        {
            System.out.println("YES");
        }
        else
        {
            System.out.println("NO");
        }
    }
}

```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```

3
10 20 30 300
10 20 170 160
179 1 179 1

```

EXPECTED OUTPUT

```

NO
YES
NO

```

Test Case 2

INPUT (STDIN)

```

4
70 40 110 140
90 20 90 160
70 40 50 200
100 30 80 150

```

EXPECTED OUTPUT

```

YES
YES
NO
YES

```

23. Question description:

Timothy is a computer science student and his teacher gave him a simple question to solve within time but he is little busy in his life, so he asked you to solve this problem.

Problem statement: You have been given an array of positive integers A_1, A_2, \dots, A_n with length N and you have to print an array of same length (N) where the values in the new array are the sum of every number in the array, except the number at that index.

Constraints:

$$1 \leq N \leq 10^5$$

$$0 \leq A[i] \leq 10^9$$

Input Format:

First line contains N , the length of the array

Second line contains N space separated positive integers.

Output Format:

Output the corresponding single array of same length

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        long sum = 0;

        StringBuilder sb = new StringBuilder();

        int n = input.nextInt();

        int[] a = new int[n];

        for (int i = 0; i < n; i++)

        {

            a[i] = input.nextInt();

            sum += a[i];

        }

        for (int i = 0; i < n; i++)

            sb.append(sum - a[i] + " ");

        System.out.println(sb);

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
4
1 2 3 4
```

EXPECTED OUTPUT

```
9 8 7 6
```

Test Case 2

INPUT (STDIN)

```
3
4 5 6
```

EXPECTED OUTPUT

```
11 10 9
```


24. Question description

Frederick opened his new store and he sells N number of items.

Price of each item is represented using an array P where P_i represents the price of item i.

A customer always buys 3 items of the lowest price from his store.

Find out the profit Frederick can make from that customer. It is guaranteed that his store has at least 3 items.

Constraints:

$$3 \leq N \leq 10^5$$

$$1 \leq A[i] \leq 10^3$$

Input Format:

First line contains N which represents the number of items in Jon's store.

Second line contains N space separated integers P1, P2, P3, ..., PN which represents the price of each item.

Output Format:

Output in a single line answer the profit Frederick makes from those 3 items.

Code:

```
import java.util.Scanner;
```

```
import java.util.Arrays;
```

```
public class Class815427160771 {  
    public static void main(String[] args) {  
        Scanner S = new Scanner(System.in);  
        int n,i,sum=0;  
        n = S.nextInt();  
        int s[] = new int[n];  
        for(i=0;i<n;i++)  
        {  
            s[i]= S.nextInt();  
        }  
        for(i=0;i<3;i++)  
        {
```

```
        Arrays.sort(s);  
        sum+=s[i];  
    }  
    System.out.println(sum);  
}  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
4 9 4 10 5	6 6 2 6 4 2 10
EXPECTED OUTPUT	EXPECTED OUTPUT
18	8

25. Question description

Lucas and his friend were scratching the lottery cards and checking whether the numbers on the screen would match

Eventually Lucas and his friend got bored so his friend gave Lucas a challenge. They considered the random numbers of lottery cards as an array of n positive integers and an integer X and task is to find the frequency of X in the array

Can you help Lucas complete the task?

Constraints:

$$1 \leq N \leq 1000$$

$$1 < A[i] \leq 10^5$$

$$1 \leq x \leq 1000$$

Input Format:

The first line contains n , the number of integers

The second line containing N integers like $A_1, A_2, A_3, \dots, A_n$

The third line contains the number X

Output Format:

The frequency or the number of times the number X has occurred in the array $A_1, A_2, A_3, \dots, A_n$.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        int n,count=0;

        int i;

        Scanner sc = new Scanner(System.in);

        n = sc.nextInt();

        int []arr=new int[n];

        for(i=0;i<n;i++)

        {

            arr[i]=sc.nextInt();

        }

        int v = sc.nextInt();

        for(i=0;i<n;i++){
```

```
        if(arr[i]==v){
            count++;
        }
    }
    System.out.println(count);
}
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
8
7 2 5 7 7 8 7 9
7
```

EXPECTED OUTPUT

```
4
```

Test Case 2

INPUT (STDIN)

```
6
1 2 4 2 3 2
2
```

EXPECTED OUTPUT

```
3
```

26. Question description

Carl has been challenged by his brother.

He has given him an array A of n integers as input. Also, you are given two integers, l and r and you have to print the subarray [l,...,r] (sub array of A from position l to position r).

If he solves this challenge his brother will give him a brand new Play Station 5? Can you help Carl and get his PS5?

Constraints:

$$1 \leq n \leq 10^6$$

$$-10^9 \leq A[i] \leq 10^9$$

Input Format:

First line will contain n, the number of elements of array.

Second line will contain array A of n integers.

Third line will contain two integers, l and r.

Output Format:

Print the subarray of A, from position l to position r.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n,i;

        int l,r;

        n = sc.nextInt();

        int []X = new int[n];

        for(i=0;i<n;i++)

        {

            X[i]=sc.nextInt();

        }

        l = sc.nextInt();

        r = sc.nextInt();

        for(i=l-1;i<r;i++)

        {
```

```
System.out.print(X[i]+" ");  
}  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
4  
2 3 -1 2  
1 3
```

EXPECTED OUTPUT

```
2 3 -1
```

Test Case 2

INPUT (STDIN)

```
6  
9 8 7 5 4 3  
1 5
```

EXPECTED OUTPUT

```
9 8 7 5 4
```

27. Question description

Hank is participating in a hackathon and he has been given an array a_1, a_2, \dots, a_n consisting of n positive integers and a positive integer m .

We have to consider two types of Sum:

1. First Sum (S_1) = $a_1 + a_2 + a_3 + \dots + a_n$
2. Second Sum (S_2) = $a_1 + a_2 + a_3 + \dots + a_m$

He has to check whether $S_1 > S_2$. If the first sum of array is greater than second sum print "YES", else print "NO".

Can you help him out?

Constraints:

$$1 \leq n \leq 10^5$$

$$1 \leq m \leq 10^5$$

$$(1 \leq a_i \leq 10^9)$$

Input Format:

The first line of each test case contains two integers n, m ($1 \leq n \leq 10^5, 1 < m < 10^5$) ($1 \leq n \leq 10^5, 1 < m \leq 10^5$).

The second line of each test case contains n integers a_1, a_2, \dots, a_n .

Output Format:

Print "YES" if $S_1 > S_2$, otherwise print "NO".

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n,m,j,s1=0,s2=0;

        n = sc.nextInt();

        m = sc.nextInt();

        int arr[] = new int[n];

        for(j=0;j<n;j++){

            arr[j]=sc.nextInt();

        }

        for(int i=0;i<n;i++){

            s1+=arr[i];
```

```
    }  
    for(int i=0;i<m;i++){  
        s2+=arr[i];  
    }  
    if(s1>s2){  
        System.out.println("YES");  
    }  
    else{  
        System.out.println("NO");  
    }  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
6 4  
98 33 58 -20 48 79
```

EXPECTED OUTPUT

```
YES
```

Test Case 2

INPUT (STDIN)

```
5 3  
1 24 9 -20 -30
```

EXPECTED OUTPUT

```
NO
```


28. Question description:

Lucas was a nice teacher, and he wanted to give a problem to the students present in class, to make them practice for the test next week.

Given an array of N integers, print the product of the numbers present in the array.

Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 10$

$0 \leq A[i] \leq 15$

Input Format:

Each test cases consists of two lines, the first line containing N, the number of elements, and the second line containing N integers.

Output Format:

Print the required answer in a single line.

Code:

```
import java.util.Scanner;
```

```
import java.util.Arrays;
```

```
public class Class815427160771 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        long ans = 1;  
        int n = sc.nextInt();  
        int arr[] = new int [n];  
        for(int i=0;i<n;i++)  
        {  
            arr[i] = sc.nextInt();  
            ans = ans*arr[i];  
        }  
        System.out.println(ans);  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
4
4 6 8 9
```

EXPECTED OUTPUT

```
1728
```

Test Case 2

INPUT (STDIN)

```
5
3 2 9 4 3
```

EXPECTED OUTPUT

```
648
```

29. Question description

Jacob has been bunking classes lately and has been caught by the guard flunking the class. His class teacher gave him an extra homework as punishment. Jacob has to finish the work quickly so

Given an array of size N consisting of only O's and I's. The array is sorted in such a manner that all the 1's are placed first and then they are followed by all the O's. Find the count of all the O's.

Constraints:

$1 < N \leq 105$

$0 \leq \text{Arr}[i] \leq 1$

Input Format:

First line contains a single integer input which specifies the number of elements in an array

The second line of input contains n integers which are either 1 or 0

Output Format:

Print the number of zeros in the array.

Code:

```
import
java.util.Scanner;
import java.util.Arrays;
public class Class815427160771 {
    public static void main(String[] args) { Scanner sc = new Scanner(System.in);
        int count=0,n,i;
        n = sc.nextInt();
        int[] arr=new int[n];
        for(i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
            if(arr[i]==0)
            {
                count++;
            }
        }
        System.out.println(count);
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
10
1 0 0 1 0 1 0 1 1 0
```

EXPECTED OUTPUT

```
5
```

Test Case 2

INPUT (STDIN)

```
5
1 0 0 1 1
```

EXPECTED OUTPUT

```
2
```

30. Question description

Davis loves travelling. Today, he will take the train from Manchester to Bristol

In this country, 24-hour clock is used. For example, 9:00 p.m. is referred to as "21 o'clock".

The current time is X o'clock, and the train will start in exactly Y hours. When will the Train Start?

Answer in 24-hour time

Constraints:

$0 \leq X, Y \leq 23$

Input Format:

Single line input consists of two integers X and Y

Output Format:

Print the hour of the starting time of the contest in 24-hour time.

Code:

```
import java.util.*;

public class Class815427160771 {
    public static class ConvertTime{
        int t;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        ConvertTime CurrentTime=new ConvertTime();
        ConvertTime RemTime=new ConvertTime();

        CurrentTime.t= sc.nextInt();
        RemTime.t= sc.nextInt();

        if(CurrentTime.t + RemTime.t >= 24)
            System.out.println((CurrentTime.t + RemTime.t) - 24);
        else
            System.out.println(CurrentTime.t + RemTime.t);
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

21 7

EXPECTED OUTPUT

4

Test Case 2

INPUT (STDIN)

11 8

EXPECTED OUTPUT

19

31. Question description

There are two rectangles. The lengths of the vertical sides of the first rectangle are U, and the lengths of the horizontal sides of the first rectangle are V. The lengths of the vertical sides of the second rectangle are X, and the lengths of the horizontal sides of the second rectangle are Y.

Print the area of the rectangle with the larger area. If the two rectangles have equal areas, print that area.

Constraints:

$$1 \leq U \leq 104$$

$$1 \leq V \leq 104$$

$$1 \leq Y \leq 104$$

Input Format:

Single line input contains 4 integers U,V,X,Y separated by space representing the lengths and breadths of first and second rectangle correspondingly

Output Format:

Print the area of the rectangle with the larger area. If the two rectangles have equal areas, print that area.

Code:

```
import java.util.*;

class Rectangle1{
    int U,V;
}

class Rectangle2{
    int X,Y;
}

public class Class815427160771 {
    public static void main(String[] args) {
        Rectangle1 R1=new Rectangle1();
        Rectangle2 R2=new Rectangle2();
        Scanner sc = new Scanner(System.in);
        R1.U= sc.nextInt();
        R1.V= sc.nextInt();
        R2.X= sc.nextInt();
        R2.Y= sc.nextInt();
    }
}
```

```
        System.out.println(Math.max(R1.U*R1.V,R2.X*R2.Y));  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

4 5 7 2

EXPECTED OUTPUT

20

Test Case 2

INPUT (STDIN)

6 9 8 9

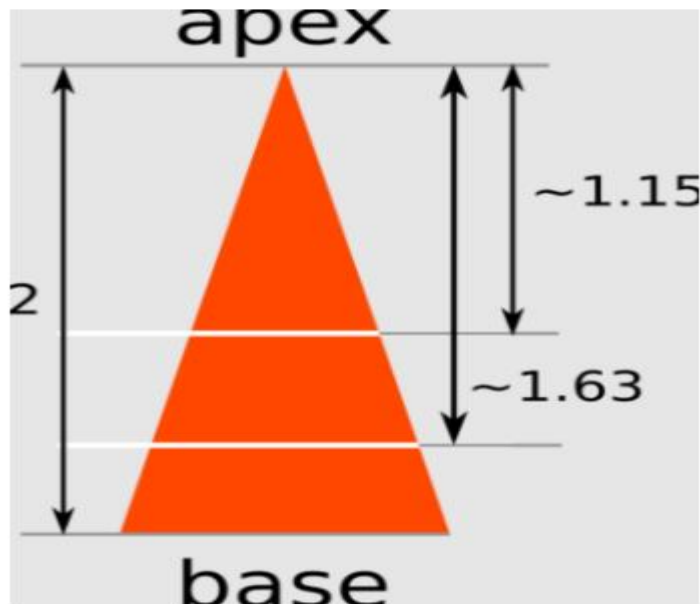
EXPECTED OUTPUT

72

32. Question description

Jax the analyst has adopted n little bunnies. As we all know, bunnies love carrots. Thus, Jax has bought a carrot to be shared between his bunnies. Jax wants to treat all the bunnies equally, and thus he wants to cut the carrot into n pieces of equal area.

Formally, the carrot can be viewed as an isosceles triangle with base length equal to 1 and height equal to h . Jax wants to make $n-1$ cuts parallel to the base to cut the carrot into n pieces. He wants to make sure that all n pieces have the same area. Can you help Jax determine where to cut the carrot so that each piece have equal area?



Constraints:

$$2 \leq n \leq 1000$$

$$1 \leq h \leq 105$$

Input Format:

The first and only line of input contains two space separated integers, n and h

Output Format:

The output should contain $n-1$ real numbers x_1, x_2, \dots, x_{n-1} . The number x_i denotes that the i th cut must be made x_i units away from the apex of the carrot.

Code:

```
import java.util.*;

class Carrot{
    int numofpieces,height;
}

public class Class815427160771{
```

```

public static void main(String args[]){
    Scanner S = new Scanner(System.in);
    Carrot divisions=new Carrot();
    divisions.numofpieces= S.nextInt();
    divisions.height= S.nextInt();
    for(int i=1;i<divisions.numofpieces;i++)
    {

        System.out.print(String.format("%.4f",divisions.height*Math.sqrt(i*1.0/divisions.numofpiec
es))+ " ");
    }
}

```

Logical Test Cases

Test Case 1

INPUT (STDIN)

3 2

EXPECTED OUTPUT

1.1547 1.6330

Test Case 2

INPUT (STDIN)

4 5

EXPECTED OUTPUT

2.5000 3.5355 4.3301

33. Question description

Kendrick has been studying about geometrical shapes recently and he thought to write a rap verse based on a geometrical shape and thought the name Pyramid sounds cool. So he wants to write a couple of lines based on the volume of Trapezoid in his song but doesn't know how to find the volume of a rectangular pyramid. Given the lengths of its length, width, and height are x, y, and z, respectively. Can you help Kendrick find the volume of Trapezoid?

Constraints:

$1 \leq x, y, z \leq 100$

Input Format:

Single line input consists of a decimal floating point value representing the lengths of its length, width, and height respectively separated by a space

Output Format:

Print the volume of the given rectangular pyramid rounded off to 4 decimal points.

Code:

```
import java.util.Scanner;

class RectPyramidArea
{
    protected float x,y,z;
}

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        RectPyramidArea area=new RectPyramidArea();

        area.x= sc.nextFloat();
        area.y= sc.nextFloat();
        area.z= sc.nextFloat();

        float a = (area.x*area.y*area.z)/3;

        System.out.println(String.format("Volume of Rectangular Pyramid: %.4f",a));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
30.4 56.9 20.1
```

EXPECTED OUTPUT

```
Volume of Rectangular  
Pyramid: 11589.3916
```

Test Case 2

INPUT (STDIN)

```
40.6 50.9 11.5
```

EXPECTED OUTPUT

```
Volume of Rectangular  
Pyramid: 7921.7368
```

34. Question description

The cab fare in Leicester is as follows:

- For the first R kilometer, the fares is P
- For the subsequent distance it is Q per kilometer

Zane would most likely take the cab today for work since he gave his car for servicing. At last he would know the total distance covered by cab.

Can you help him to calculate the total fare for the whole ride?

Constraints:

$0 \leq R, P, Q, D \leq 105$

Input Format:

Single line input contains four integers representing the first R kilometers followed by P which is the corresponding fare for first R kilometers followed by Q which is the fare for the remaining distance covered and D which is the total distance travelled and all of them are separated by a space.

Output Format:

Print the value of total fare

Code:

```
import java.util.*;

public class Class815427160771 {

    protected static class InitialFare{

        int R,P;

    }

    protected static class SubsequentFare{

        int Q,D;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        InitialFare IF= new InitialFare();

        SubsequentFare SF= new SubsequentFare();

        IF.R= sc.nextInt();

        IF.P= sc.nextInt();

        SF.Q= sc.nextInt();

        SF.D= sc.nextInt();

    }

}
```

```
        System.out.println(IF.R*IF.P+(SF.D-IF.R)*SF.Q);  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

15 20 40 25

EXPECTED OUTPUT

700

Test Case 2

INPUT (STDIN)

20 68 32 70

EXPECTED OUTPUT

2960

35. Question description

Brody went across the neighbourhood to get some lemonade and he came across a lemonade stand hosting a small contest and if the person who takes the challenge clears it up they get a glass of lemonade for free. The statement of the contest was, given non negative integers u and v ($u \leq v$), and a positive integer w . Among the integers between u and v , inclusive, how many are divisible by w ?

Can you help Brody get a nice cool glass of lemonade for free?

Constraints: $1 \leq u, v \leq 10^8$

$1 \leq w \leq 10^8$

Input Format:

Single line input contains three integers u , v and w separated by space

Output Format:

Print the number of the integers between u and v , inclusive, that are divisible by w .

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    private static class InputNumbers
    {
        protected int a,b,c;
    }

    public static void main(String[] args) {
        InputNumbers obj=new InputNumbers();

        Scanner sc = new Scanner(System.in);

        obj.a= sc.nextInt();
        obj.b= sc.nextInt();
        obj.c= sc.nextInt();

        int count=0;
        for(int i=obj.a;i<=obj.b;i++){
            if(i%obj.c==0){
                count++;
            }
        }
    }
}
```

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

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
10 100 4	8 30 4
EXPECTED OUTPUT	EXPECTED OUTPUT
23	6

36. Question description

There are three sticks with integer lengths 11,12,13

You are asked to break exactly one of them into two pieces in such a way that:

- both pieces have positive (strictly greater than 0) integer length;
- the total length of the pieces is equal to the original length of the stick;
- it's possible to construct a rectangle from the resulting four sticks such that each stick is used as exactly one of its sides.

A square is also considered a rectangle.

Determine if it's possible to do that.

Constraints:

Input Format:

The only line of each testcase contains three integers 11,12,13 the lengths of the sticks.

Output Format:

Print "POSSIBLE" if it's possible to break one of the sticks into two pieces with positive integer length in such a way that it's possible to construct a rectangle from the resulting four sticks. Otherwise, print "NOT POSSIBLE".

Code:

```
import java.util.Scanner;

class StickRectangles
{
    int s1,s2,s3;
}

public class Class815427160771 {

    public static void main(String[] args) {

        StickRectangles stick=new StickRectangles();

        Scanner S = new Scanner(System.in);

        stick.s1= S.nextInt();

        stick.s2= S.nextInt();

        stick.s3= S.nextInt();

        if(stick.s1 == stick.s2+stick.s3 || stick.s2 == stick.s1+stick.s3 || stick.s3 ==
        stick.s1+stick.s2)

        {

            System.out.println("POSSIBLE");
```

```
    }  
    else  
    {  
        System.out.println("NOT POSSIBLE");  
    }  
}  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
6 1 5	2 5 2
EXPECTED OUTPUT	EXPECTED OUTPUT
POSSIBLE	NOT POSSIBLE

37. Question description

Two students of Berkshire Kindergarten are fighting over candy packs.

There are three candy packs, each of which contains a, b, and c candies, respectively.

Teacher Mary is trying to distribute the packs between the two students so that each student gets the same number of candies. Determine whether it is possible.

Note that Mary cannot take candies out of the packs, and the whole contents of each pack must be given to one of the students.

Constraints:

$1 \leq a, b, c \leq 1000$

Input Format:

Single line of input, with three space-separated positive integers a, b and c.

Output Format:

If it is possible to distribute the packs so that each student gets the same number of candies, print Yes. Otherwise, print No.

Code:

```
import java.util.Scanner;

class ClassAdmin {
    public static class Pattern
    {
        int a,b,c;
    }
}

public class Class815427160771 {
    public static void main(String[] args) {
        ClassAdmin.Pattern obj=new ClassAdmin.Pattern();
        Scanner sc = new Scanner(System.in);
        obj.a= sc.nextInt();
        obj.b= sc.nextInt();
        obj.c= sc.nextInt();
        if(obj.c==obj.a+obj.b || obj.a==obj.b+obj.c || obj.b==obj.a+obj.c)
            System.out.println("YES");
        else
```

```
        System.out.println("NO");  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

10 30 20

EXPECTED OUTPUT

YES

Test Case 2

INPUT (STDIN)

30 30 100

EXPECTED OUTPUT

NO

38. Question description

Brady is given a list of N triangles. Each triangle is identified by the coordinates of its three corners in the 2-D cartesian plane. His job is to figure out how many of the given triangles are right triangles. A right triangle is a triangle in which one angle is a 90 degree angle. The vertices of the triangles have integer coordinates and all the triangles given are valid(three points aren't colinear).

Constraints:

$1 \leq X1, Y1, X2, Y2, X3, Y3 < 20$

Input Format:

Single line input contains six space separated integers X1,Y1,X2,Y2,X3, Y3 where (X1,Y1) (X2,Y2) (X3,Y3) are the vertices of a triangle.

Output Format:

Print "It is a Right Angled Triangle" if the given coordinates form a right angle triangle otherwise print "Not a Right Angled Triangle".

Code:

```
import java.util.Scanner;

class Coordinates
{
    public static class FirstCoordinate{
        int X1,Y1; }

    public static class SecondCoordinate{
        int X2,Y2; }

    public static class ThirdCoordinate{
        int X3,Y3; }
}

public class Class815427160771{

    public static void main (String[] args)    {

        Coordinates.FirstCoordinate point1=new Coordinates.FirstCoordinate();
        Coordinates.SecondCoordinate point2=new Coordinates.SecondCoordinate();
        Coordinates.ThirdCoordinate point3=new Coordinates.ThirdCoordinate();

        Scanner Sc = new Scanner(System.in);

        point1.X1= Sc.nextInt();
        point1.Y1= Sc.nextInt();
        point2.X2= Sc.nextInt();
```

```

        point2.Y2= Sc.nextInt();

        point3.X3= Sc.nextInt();

        point3.Y3= Sc.nextInt();


        int A = (int)Math.pow((point2.X2 - point1.X1), 2) +(int)Math.pow((point2.Y2
- point1.Y1), 2);

        int B = (int)Math.pow((point3.X3 - point2.X2), 2) +(int)Math.pow((point3.Y3 - point2.Y2), 2);
        int C = (int)Math.pow((point3.X3 - point1.X1), 2) +(int)Math.pow((point3.Y3 - point1.Y1), 2);
        // Check Pythagoras Formula
        if ((A == (B + C) || B == (A + C) || C == (A + B)))
        {
            System.out.println("It is a Right Angled Triangle");
        }
        else
        {
            System.out.println("Not a Right Angled Triangle");
        }
    }
}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
0 2 0 14 9 2	17 19 12 16 19 0
EXPECTED OUTPUT	EXPECTED OUTPUT
It is a Right Angled Triangle	Not a Right Angled Triangle

39. Question description

Neil has planned a trip using trains and buses. The train fare will be M euros if you buy ordinary tickets along the way, and Neuros if you buy an unlimited ticket. Similarly, the bus fare will be O euros if you buy ordinary tickets along the way, and Peuros if you buy an unlimited ticket.

Find the minimum total fare when the optimal choices are made for trains and buses.

Constraints:

$1 < M < 1000$

$1 < N \leq 1000$

$1 \leq O \leq 1000$

$1 < P \leq 1000$

Input Format:

Single line input contains four integers M,N,O,P separated by a space

Output Format:

Print the minimum total fare which can help Neil save some money.

Code:

```
import java.util.Scanner;

class TrainFare
{
    int M,N;
}

class BusFare
{
    int O,P;
}

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        TrainFare train=new TrainFare();

        BusFare bus=new BusFare();

        train.M= sc.nextInt();

        train.N= sc.nextInt();

        bus.O= sc.nextInt();
```

```
        bus.P= sc.nextInt();  
        System.out.println(Math.min(train.M, train.N) + Math.min(bus.O, bus.P));  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

600 300 220 420

EXPECTED OUTPUT

520

Test Case 2

INPUT (STDIN)

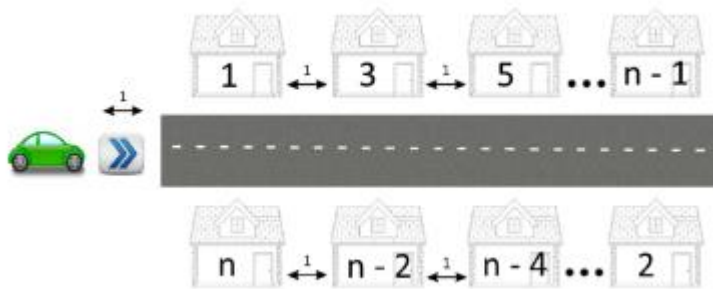
445 625 655 790

EXPECTED OUTPUT

1100

40. Question description

The main street of Candyland is a straight line with n houses built along it (n is an even number). The houses are located at both sides of the street. The houses with odd numbers are at one side of the street and are numbered from 1 to $n-1$ in the order from the beginning of the street to the end (in the picture: from left to right). The houses with even numbers are at the other side of the street and are numbered from 2 to n in the order from the end of the street to its beginning (in the picture: from right to left). The corresponding houses with even and odd numbers are strictly opposite each other, that is, house 1 is opposite house n , house 3 is opposite house $n-2$, house 5 is opposite house $n-4$ and so on.



Brad needs to get to house number a as quickly as possible. He starts driving from the beginning of the street and drives his car to house a . To get from the beginning of the street to houses number 1 and n , he spends exactly 1 second. He also spends exactly one second to drive the distance between two neighbouring houses. Brad can park at any side of the road, so the distance between the beginning of the street at the houses that stand opposite one another should be considered the same.

Your task is: find the minimum time Brad needs to reach house a .

Constraints:

$1 \leq a \leq n \leq 100\,000$

Input Format:

The first line of the input contains two integers, n and a - the number of houses on the street and the number of the house that Brad needs to reach, correspondingly. It is guaranteed that number n is even.

Output Format:

Print a single integer — the minimum time Brad needs to get from the beginning of the street to house a .

Code:

```
import java.util.*;
```

```
public class Class815427160771 {
```

```
    public static void housenumbers(int x,int y) {
```

```
        if(y%2==0) {
```

```

        System.out.print((x-y)/2+1);
    }
    else {
        System.out.print(y/2+1);
    }
}

public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int n,a;
    n= sc.nextInt();
    a= sc.nextInt();
    housenumbers(n,a);
}
}

```

Logical Test Cases

Test Case 1

INPUT (STDIN)

1628 274

EXPECTED OUTPUT

678

Test Case 2

INPUT (STDIN)

1764 1288

EXPECTED OUTPUT

239

41. Question description

Donald seemingly down-to-earth guy. Donald has a Brother Ronald. Ronald leads a life of a computer hacker by day and a thief by night. One day, Ronald tries to break the door digital lock of Donald's room.

Donald who wants to prevent it thinks to give Passcode for security.

But it only accepts a perfect number as input. Can you help him to make a program for checking the given number is the perfect number or not...

Explanation:

For example, 6 is a positive number that is completely divisible by 1, 2, and 3.

We know that the number is also divisible by itself but we will include it in the addition of divisors. When we add these divisors ($1 + 2 + 3 = 6$), it produces 6, which is equal to the number that we have considered. So, we can say that 6 is a perfect number.

Constraints:

$1 \leq \text{num} \leq 10000$

Input Format:

Single line input contains an integer num

Output Format:

Print whether the given number is a perfect number or not a perfect number.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static int perfect(int num){

        int rem, sum = 0, i;

        for(i=1;i<=num/2;i++)

        {

            rem = num % i;

            if (rem == 0){

                sum = sum + i;

            }

        }

        if (sum == num)

            System.out.println("Perfect Number");

        else
```

```

        System.out.println("Not a Perfect Number");
        return sum;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a= sc.nextInt();
        perfect(a);
    }
}

```

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
496	10
EXPECTED OUTPUT	EXPECTED OUTPUT
Perfect Number	Not a Perfect Number

42. Question description

Alfred likes everything infinite. Now he is studying the properties of a sequence s , such that its first element is equal to a ($s_1 = a$), and the difference between any two neighbouring elements is equal to c ($s_i - s_{i-1} = c$). In particular, Alfred wonders if his favourite integer b appears in this sequence, that is, there exists a positive integer i , such that $s_i = b$. Of course, you are the person he asks for a help.

Constraints:

$-109 \leq a, b, c \leq 109$

Input Format:

The first line of the input contain three integers a , b and c - the first element of the sequence, Alfred's favorite number and the difference between any two neighbouring elements of the sequence, respectively.

Output Format:

If b appears in the sequence s print "YES" (without quotes), otherwise print "NO" (without quotes).

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void infsequence(int x,int y,int z){

        if(z==0) {

            if(x==y){

                System.out.println("YES");

            }

            else {

                System.out.println("NO");

            }

            return;}

        if((y-x)%z==0 && (y-x)/z>=0){

            System.out.println("YES");

        }

        else {

            System.out.println("NO");

        }

    }

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);
```

```
int a,b,c;  
a= sc.nextInt();  
b= sc.nextInt();  
c= sc.nextInt();  
    infsequence(a,b,c);  
}}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1 7 3

EXPECTED OUTPUT

YES

Test Case 2

INPUT (STDIN)

-2 -1 2

EXPECTED OUTPUT

NO

43. Question description

Travis Scott shared pictures of his batman suit on Instagram but it ended up becoming a meme . Felix is hosting meme review this Friday and Therefore, he decided to take a string S and change each of its substrings that spells "Travis" to "Batman". Find the resulting string.

Constraints:

$$1 < |S| \leq 105$$

Input Format:

The first and only line of each test case contains a single string S

Output Format:

Print a single line containing the string after it is modified by Felix by replacing "Travis" with "Batman"

Code:

```
import java.util.Scanner;

import java.lang.String;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String s;

        s=sc.nextLine();

        s = s.replaceAll( "Travis" , "Batman" );

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

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
Travis Scott meal is
awesome
```

EXPECTED OUTPUT

```
Batman Scott meal is
awesome
```

Test Case 2

INPUT (STDIN)

```
Astro world album by
Travis was cool
```

EXPECTED OUTPUT

```
Astro world album by
Batman was cool
```

44. Question description

Nicholas take the metro everyday from Brooklyn to Manhattan and likes to read the newspaper during the journey. He eventually got bored of reading the routine news everyday so he started playing a small game to keep himself entertained. He started to count the number of alphabets, digits and special characters in a whole sentence of the newspaper.

Can you also join Nicholas and hang out with him?

Constraints:

Input Format:

Single line input containing the String S which is a complete sentence

Output Format:

Print the number of alphabets, digits and special characters in the sentence.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner S = new Scanner(System.in);

        int ascii;

        int alphabets=0,digits=0,splchar=0;

        String st= S.nextLine();

        char[]str=st.toCharArray();

        for(int i=0;i<str.length;i++)

        {

            ascii = str[i];

            if(ascii >= 48 && ascii <= 57 ) {

                digits++;

            }

            else if(ascii >= 65 && ascii <= 90 || ascii >= 97 && ascii <= 122) {

                alphabets++;

            }

            else {

                splchar++;

            }

        }

    }

}
```



```

        }

    }

    System.out.println("Number of Alphabets: " + alphabets);

    System.out.println("Number of Digits: " + digits);

    System.out.println("Number of Special Characters: " + splchar);

}

}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
<p>INPUT (STDIN)</p> <pre>www.trivago.com/hotels/9989</pre>	<p>INPUT (STDIN)</p> <pre>www.facebook.com/stocks/11789\$sign</pre>
<p>EXPECTED OUTPUT</p> <pre>Number of Alphabets: 19 Number of Digits: 4 Number of Special Characters: 4</pre>	<p>EXPECTED OUTPUT</p> <pre>Number of Alphabets: 24 Number of Digits: 5 Number of Special Characters: 5</pre>

45. Question description

Mike is getting really bored during the lecture so he started to add all the digits of the current page number the teacher is reading from the textbook and called the sum $S(n)$

Let $S(n)$ denote the sum of the digits in the decimal notation of the page number. For example, $S(101)=1+0+1=2$.

Given an integer string N , determine if $S(M)$ divides N .

Constraints

Input Format:

Single line input which has a string S which consists of a number representing the page number

Output Format:

If $S(M)$ divides N , print Yes; if it does not, print No.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String str;

        int sum=0;

        str = sc.next();

        int num = Integer.valueOf(str);

        String number = String.valueOf(num);

        for (int i=0;i<str.length();i++){

            int j = Character.digit(number.charAt(i),10);

            sum+=j;

        }

        if(num%sum==0){

            System.out.println("Yes");

        }

        else{

            System.out.println("No");

        }

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

18

EXPECTED OUTPUT

Yes

Test Case 2

INPUT (STDIN)

97

EXPECTED OUTPUT

No

46. Question description

Hamilton and Jake are preparing themselves for programming contests.

To train their logical thinking and solve problems better, Hamilton and Jake decided to play chess. During the game Hamilton wondered whose position is now stronger.

For each chess piece we know its weight:

- the queen's weight is 9,
- the rook's weight is 5,
- the bishop's weight is 3,
- the knight's weight is 3,
- the pawn's weight is 1,
- the king's weight isn't considered in evaluating position.

The player's weight equals to the sum of weights of all his pieces on the board.

As Hamilton doesn't like counting, he asked you to help him determine which player has the larger position weight.

Constraints:

Input Format:

The input contains eight lines, eight characters each - the board's description.

The white pieces on the board are marked with uppercase letters, the black pieces are marked with lowercase letters. The white pieces are denoted as follows: the queen is represented is 'Q', the rook - as 'R', the bishop - as 'B', the knight - as 'N', the pawn - as 'P', the king - as 'K'.

The black pieces are denoted as 'q', 'r', 'b', 'n', 'p', 'k', respectively.

An empty square of the board is marked as '.' (a dot).

It is not guaranteed that the given chess position can be achieved in a real game. Specifically, there can be an arbitrary (possibly zero) number pieces of each type, the king may be under attack and so on.

Output Format:

Print "White" (without quotes) if the weight of the position of the white pieces is more than the weight of the position of the black pieces, print "Black" if the weight of the black pieces is more than the weight of the white pieces and print "Draw" if the weights of the white and black pieces are equal.

Code:

```
import java.util.Scanner;

interface Chess{

    String []s=new String[1000000];
```

```

        void gameon(String [] s);
    }
class Moves implements Chess{
    public void gameon(String []s){
        Scanner input = new Scanner(System.in);

        char[] white ={ 'Q', 'R', 'B', 'N', 'P', 'K' };
        char[] black ={ 'q', 'r', 'b', 'n', 'p', 'k'};
        int[] verif={ 9, 5, 3, 3, 1, 0};
        int s_white=0,s_black=0;

        for (int i = 0; i < 8; i++) {
            s[i] =input.nextLine();
            for(int j=0; j<s[i].length(); j++) {
                for(int k=0; k<6; k++) {
                    if(s[i].charAt(j)==white[k]) {
                        s_white+=verif[k];
                    }
                    if(s[i].charAt(j)==black[k]) {
                        s_black+=verif[k];}}}

        }

        if(s_white>s_black)
            System.out.print("White");
        if(s_white<s_black)
            System.out.print("Black");
        if(s_white==s_black)
            System.out.print("Draw");
        input.close();
    }
}

public class Class815427160771 {
    public static void main(String[] args) {
        Moves obj=new Moves();
    }
}

```

```
        obj.gameon(obj.s);  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
...QK...  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
...rk...
```

EXPECTED OUTPUT

White

Test Case 2

INPUT (STDIN)

```
rnbqkbnr  
pppppppp  
.....  
.....  
.....  
.....  
.....  
PPPPPPPP  
RNBQKBNR
```

EXPECTED OUTPUT

Draw

47. Question description

Collin, Cindy, Hannah, and Hunter are four friends sharing a one-room workspace. The workspace has a single thermostat which they can set to any integer temperature between 35 degrees to 95 degrees Fahrenheit, inclusive.

The four friends can't agree on the room's temperature! Collin and Cindy don't want it to be too cold, while Hannah and Hunter don't want it to be too hot. Specifically:

- Collin wants it to be at least c_1 degrees Fahrenheit.

Cindy wants it to be at least c_2 degrees Fahrenheit.

- Hannah wants it to be at most h_1 degrees Fahrenheit.

Hunter wants it to be at most h_2 degrees Fahrenheit.

Given c_1 , c_2 , h_1 and h_2 and is there a satisfactory temperature that all four friends will be happy with? If it's possible, print YES; otherwise, print NO.

Constraints:

$1 \leq c_1, c_2, h_1, h_2 \leq 95$

Input Format:

Four space-separated integers describing the respective values of c_1 , c_2 , h_1 and h_2

Output Format:

Print YES if it's possible to satisfy all four friends' conditions; otherwise, print NO instead.

Code:

```
import java.util.*;

class Temperature{
    int c1,c2,h1,h2;
}

class Fahrenheit extends Temperature{
    String isSatisfiable(int c1,int c2,int h1,int h2){
        if(Math.max(c1,c2)<=Math.min(h1,h2)){
            return "YES";
        }
        else {
            return "NO";
        }
    }
}
```

```
}
```

```
public class Class815427160771 {  
    public static void main(String[] args) {  
        Temperature temp=new Temperature();  
        Fahrenheit burn=new Fahrenheit();  
        Scanner sc = new Scanner(System.in);  
        temp.c1=sc.nextInt();  
        temp.c2=sc.nextInt();  
        temp.h1=sc.nextInt();  
        temp.h2=sc.nextInt();  
        String result = burn.isSatisfiable(temp.c1,temp.c2,temp.h1,temp.h2);  
        System.out.println(result);  
    }  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
50 40 70 60	20 40 10 60
EXPECTED OUTPUT	EXPECTED OUTPUT
YES	NO

48. Problem Description:

Karthik was working in the HR division of Audi.

The employees of the company were working on shifts.

The company calculates salary for the employees on the basis of employee working hours per day.

Since the number of people working in the company is huge salary calculation become a tedious process at the end of the each day.

Constraints:

$1 \leq \text{hour} \leq 12$

$1 \leq \text{salaryperday} \leq 6000$

Input Format:

The First line of the input has a single value representing the total working hours of type integer.

The Second line of the input has single value representing the salary per day of type double.

Output Format:

Print the total salary in single line with two values after decimal point.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int hour = input.nextInt();

        double salaryperday = input.nextDouble();

        double tot = hour*salaryperday;

        System.out.println(String.format("%.2f",tot));

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

7
2945.89

EXPECTED OUTPUT

20621.23

Test Case 2

INPUT (STDIN)

11
5781.56

EXPECTED OUTPUT

63597.16

49. Problem Description:

Swathy and Nancy were selected for Spacey programme which was about to take place the next year, in their interview they were struck with the question.

The question is that if the floating number is given they have to create a code to display the rightmost integer from the integer part of the number.

If they have the logic for the code they will be the part of the digital meter designing for the Spacey Mars launch which was their dream.

Can you help them with a logic of the code for the criteria given to them?

Constraints:

$25.0000 \leq \text{spacenum} < 999.0000$

Input Format:

Only Line of Input has single value of type float. Output Format:

Print the rightmost integer from the input value.

Explanation:

If the input is given 124.34, then the output to be displayed is 4 (i.e) Before decimal the integral part is 124, in that last digit is 4.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int num, digit;

        float spacenum = input.nextFloat();

        num=(int)spacenum;

        digit = num%10;

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

✓ Logical Test Cases

Test Case 1
INPUT (STDIN)
345.476
EXPECTED OUTPUT
5

Test Case 2
INPUT (STDIN)
759.231
EXPECTED OUTPUT
9

50. Problem Description:

Arul and Kani own the farm in the beautiful location of the city where lot of cows were roaming around.

One day Arul and Kani were out of the city.

On that day cows have eaten the grasses in the farm which is circular in structure. When Arul and Kani reached the location they were shocked to see the grass being eaten by cows.

Now they would like to know for how much area and circumference of the farm the cows have eaten the grass.

Can you help them find it.

Functional Description:

Circumference = $2 * \pi * r$

Area = $\pi * r * r$

$\pi = 3.14$

Constraints:

$1.00 \leq \text{rad} \leq 100.00$

Input Format:

The only line of the input represents the radius of the circle of type float.

Output Format:

Print the area in the first line and circumference in the second line with only 2 values after decimal point

Code:

```
import java.util.Scanner;
```

```
public class Class815427160771 {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        double rad = sc.nextDouble();
```

```
        double area = Math.PI * Math.pow(rad, 2);
```

```
        double circumference = (2*Math.PI*rad);
```

```
        System.out.println(String.format("%.2f", area));
```

```
        System.out.println(String.format("%.2f", circumference));
```

```
    }
```

```
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

78.6

EXPECTED OUTPUT

19408.63

493.86

Test Case 2

INPUT (STDIN)

91.3

EXPECTED OUTPUT

26187.34

573.65

51. Problem Description:

Flipkart announced the year end stock clearance sale and as apart of they have also conducting the contest and the users answering the questions asked in the contest can win Moto One Power free of cost.

The task is to display the first three powers (N^1 , N^2 , N^3) of the given.

Nishanth was looking to buy Moto One Power.

If you help nishanth in solving the task he will get his favorite mobile.Can you help him?

Constraints:

$1 \leq N \leq 150$

Input Format:

Only line of the input has a single integer representing N.

Output Format:

Print the first three powers of N in a single line separated by a space.

Code:

```
import java.lang.Math;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int N = input.nextInt();

        double NP = (double)N;

        System.out.print(N);

        System.out.print(String.format(" %.0f", (Math.pow(NP, 2))));

        System.out.print(String.format(" %.0f", (Math.pow(NP, 3))));

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

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

137

EXPECTED OUTPUT

137 18769 2571353

Test Case 2

INPUT (STDIN)

98

EXPECTED OUTPUT

98 9604 941192

52. Problem Description:

Sajid loves super heroes he used to imagine himself to be a hero.

One day his teacher was taking a class about shapes and asked the students to find the Area of the triangle using Heron's formula.

Sajid misheard this as Hero's formula and was curious to discover the Hero's formula for finding the area of the triangle.

Help Sajid to solve his math problem by using the correct logic in your code.

Functional Description:

Area = $\sqrt{s(sa)(s-b)(sc)}$, where

$s = (a+b+c)/2$

and

a, b & c are the sides of triangle.

Constraints:

1 ≤ a ≤ 15

1 ≤ b ≤ 15

1 ≤ c ≤ 15

Input Format:

Only Line of input has 3 integers representing 3 side of the triangle separated by a space.

Output Format:

Print the area of the triangle with only two values after the decimal point

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int a=input.nextInt();
        int b=input.nextInt();
        int c=input.nextInt();
```

```
        double s = (a+b+c)/2;  
        double area = s*(s-a)*(s-b)*(s-c);  
        System.out.println(String.format("%.2f",(Math.sqrt(area))));  
    }  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
5 5 4	6 5 4
EXPECTED OUTPUT	EXPECTED OUTPUT
9.17	6.48

53. Problem Description:

Roopa and Atifa are sisters they love to compete by playing math games which gradually helped them in their academics one day.

Roopa gave her a math puzzle to her sister.

The puzzle involves two decimal numbers.

Atifa just had add the two floating point numbers but the twist is Atifa should only add the integer part of the decimal number.

But Atifa thought she can code a program for the puzzle but she is finding it difficult.

Can you help her with the suitable logic?

Explanation:

Roopa gave her sister two floating point numbers: 23.44 and 33.22 Her sister need to identify the left side integral part of given numbers are 23 and 33. Finally the added output value is 56.

Constraint:

$0.00 < \text{num1} < 500.00$

$0.00 < \text{num2} < 500.00$

Input Format:

First Line: Has Single Input of type float representing first number

Second Line: Has Single Input of type float representing second number

Output Format:

Print the Sum of Integer parts of two numbers.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        float num1=input.nextFloat();
        float num2=input.nextFloat();
        float sum = num1 + num2;
        int sum1 = (int)sum;
        System.out.println(Math.round(sum1));
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

99.26

34.46

EXPECTED OUTPUT

133

Test Case 2

INPUT (STDIN)

139.25

163.18

EXPECTED OUTPUT

302

54. Problem Description:

Arif planned to make a room cleaning robot for his college mini project competition.

First he has to code program to simulate the robot movements inside the room.

He measured the length and width of the room.

Once the values are available, his program should compute and display the area of the room.

Can you help Arif with a suitable logic for the code?

Constraint:

$20.00 \leq \text{length} < 100.00$

$20.00 < \text{width} \leq 100.00$

Input format:

First Line:has single floating point number representing length of the room Second Line:has single floating point number representing width of the room

Output format:

Print the area of the room in square feet.

Code:

```
import java.io.*;
```

```
import java.util.Scanner;
```

```
public class Class815427160771 {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        float length = input.nextFloat();
```

```
        float width = input.nextFloat();
```

```
        System.out.println(String.format("%.2f", (length*width))+" sq.ft");
```

```
    } }
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
23.44 22.33	27.42 29.35
EXPECTED OUTPUT	EXPECTED OUTPUT
523.42 sq.ft	804.78 sq.ft

55. Problem Description:

Issac loved to do agriculture he worked for a 9-5 job in the week days and dedicated to do agriculture on the week end.

He dreamed to combine technology and agriculture together in the future. He started with a small automated automobile that can water the plants when he is not available in the field.

He measured his field in square feet but for generalising his project he wished to convert it to acres.

Can you help him with a code that reads the area of the farmer's field in square feet and display the area in acres?

Functional Description:

There are 43,560 square feet in an acre.

Constraints:

$20000.00 \leq \text{tractLand} \leq 70000.00$

Input format:

Single Line of Input has a tractland's area in square feet of type float.

Output format:

Print the input area of the tractLand in square feet and its equivalent area in acres in a single line.

Refer sample testcases for formating information.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        float tractLand, tractLandAcred;
        tractLand = input.nextFloat();
        tractLandAcred = tractLand/43560;
        System.out.println((tractLand) + " sq.ft is equal to " +
        (String.format("%.2f", tractLandAcred)) + " acres");
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
45587.23
```

EXPECTED OUTPUT

```
45587.23 sq.ft is equal  
to 1.05 acres
```

Test Case 2

INPUT (STDIN)

```
58887.23
```

EXPECTED OUTPUT

```
58887.23 sq.ft is equal  
to 1.35 acres
```

56. Problem Description:

Aaron took his girl friend Binita to a restaurant as he got a job of his dreams. Since he had small welcome interview he was little bit tensed.

Binita figured this out and to get back Aaron's confidence she gave him a little task,

When they received the bill for the food they ordered, she asked aaron to find out the tax amount of the bill and tip for the meal through a computer code.

Aaron can use your local tax rate when computing the amount of tax owing.

Can you help aaron to code the suitable logic.

Note:

Local tax= 18%

Tip amount=5%

Constraint:

50 ≤ billwt ≤ 1300

Where billwt is the variable you should use for getting the bill amount without tax and tip.

Input format:

Single Line of input has single value of type integer representing the Bill Amount Without Tax and Tip

Output format:

In First Line Print the calculated Tax with only 2 values after decimal point.

In the Second Line Print the calculated Tip with only 2 values after decimal point.

In the Third Line Print the Total Bill Amount with tax and tip with only 2 values after decimal point.

Code:

```
import java.io.*;
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int billwt;
        billwt = input.nextInt();
        double b = (double)billwt;
        double Tax = (b*18/100);
        double Tip = (b*5/100);
        double Total = Tax + Tip + b;
```

```
        System.out.println("The Tax is "+ (String.format("%.2f",Tax)));  
        System.out.println("The Tip is "+ (String.format("%.2f",Tip)));  
        System.out.println("Total Bill With Tax and Tip is "+ (String.format("%.2f",Total)));  
    }  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
521	734
EXPECTED OUTPUT	EXPECTED OUTPUT
The Tax is 93.78 The Tip is 26.05 Total Bill With Tax and Tip is 640.83	The Tax is 132.12 The Tip is 36.70 Total Bill With Tax and Tip is 902.82

57. Problem Description:

Nathan was a student by morning and a computer nerd by night.

At the earlier stages of his career he was in need of money,

So he started working in a grocery store. In the grocery store he need to get the product ID, price of the product(Price per Unit) and the quantity of the product purchased by the customer.

At point of time he found he was doing the same job again and again so he thought of automating the task.

Help Nathan for framing the code for his work.

Constraint:

$1000 < \text{billid} < 9999$

$1000 \leq \text{prodid} \leq 9999$

$10.00 \leq \text{price} < 500.00$

$1 \leq \text{quantity} < 500$

Input Format:

First line has the bill id in integer format

Second line has the product id in integer format

Third line contains the product's price in float format

Fourth line contains the quantity of purchased items in integer format

Output Format:

Print the bill amount corresponding to the bill id.

Code:

```
import java.io.*;

import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int prodid,billid,quantity;

        float price,totprice;

        billid = input.nextInt();

        prodid = input.nextInt();

        price = input.nextFloat();

        quantity = input.nextInt();
```



```
        totprice = price * quantity;  
        System.out.println(String.format("%.2f",totprice));  
    }  
}
```

▼ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
1521 1113 78.00 4	1522 1034 39.50 17
EXPECTED OUTPUT	EXPECTED OUTPUT
312.00	671.50

58. Question description

Pete and his friends are walking along the fence of height h and they do not want the guard to notice them. In order to achieve this the height of each of the friends should not exceed h . If the height of some person is greater than h he can bend down and then he surely won't be noticed by the guard. The height of the i th person is equal to

Consider the width of the person walking as usual to be equal to 1, while the width of the bent person is equal to 2. Friends want to talk to each other while walking, so they would like to walk in a single row. What is the minimum width of the road, such that friends can walk in a row and remain unattended by the guard?

Constraints:

$$1 \leq n \leq 1000$$

$$1 \leq h \leq 1000$$

Input Format:

The first line of the input contains two integers n and h — the number of friends and the height of the fence, respectively.

The second line contains n integers a_i , the i th of them is equal to the height of the i -th person.

Output Format:

Print a single integer - the minimum possible valid width of the road.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        int n,m;

        n=sc.nextInt();

        m=sc.nextInt();

        int w=0;

        for(int i=0;i<n;i++){

            if(m<=sc.nextInt())w++;

            else w+=2;

        }

        System.out.println(w);

    } }
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3 7
4 5 14
```

EXPECTED OUTPUT

```
4
```

Test Case 2

INPUT (STDIN)

```
6 5
7 6 8 9 10 5
```

EXPECTED OUTPUT

```
11
```

59. Question description

Let's consider a table consisting of n rows and n columns. The cell located at the intersection of i th row and i th column contains number ixj . The rows and columns are numbered starting from 1.

You are given a positive integer x . Your task is to count the number of cells in a table that contain number x .

Constraints:

$$1 \leq n \leq 105$$

$$1 \leq x \leq 109$$

Input Format:

The single line contains numbers n and x we are looking for in the table. -the size of the table and the number that

Output Format:

Print a single number: the number of times x occurs in the table.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        int n,x;

        n=sc.nextInt();

        x=sc.nextInt();

        int c=0;

        for (int i = 1; i <= n; i++)

            if (x % i == 0 && x / i <= n) c++;

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

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

10 5

EXPECTED OUTPUT

2

Test Case 2

INPUT (STDIN)

6 12

EXPECTED OUTPUT

4

60. Question description

Joyner is in love with lovely numbers! A positive integer x is called lovely if and only if there is no such positive integer $a > 1$ such that a^2 is a divisor of x .

Malek has a number store! In his store, he has only divisors of positive integer n (and he has all of them). As a birthday present, Malek wants to give her a lovely number from his store. He wants this number to be as big as possible.

Malek always had issues in math, so he asked for your help. Please tell him what is the biggest lovely number in his store.

Constraints:

$1 \leq n \leq 10^{12}$

Input Format:

The first and only line of input contains one integer, n .

Output Format:

Print the answer in one line.

Code:

```
import java.util.*;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        long n=sc.nextLong();
        for(long i=2;i*i<=n;i++){
            while(n%(i*i)==0){
                n=n/i;
            }
        }
        System.out.println(n);
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

12

EXPECTED OUTPUT

6

Test Case 2

INPUT (STDIN)

2457

EXPECTED OUTPUT

273

61. Question description

Consider the infinite sequence of integers: 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5.... The sequence is built in the following way: at first the number 1 is written out, then the numbers from 1 to 2, then the numbers from 1 to 3, then the numbers from 1 to 4 and so on. Note that the sequence contains numbers, not digits. For example number 10 first appears in the sequence in position 55 (the elements are numerated from one).

Find the number on the n-th position of the sequence.

Constraints:

$1 \leq n \leq 10^{14}$

Input Format:

The only line contains integer n - the position of the number to find.

Output Format:

Print the element in the n-th position of the sequence (the elements are numerated from one).

Code:

```
import java.util.*;

public class Class815427160771{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        long n = sc.nextLong();

        long x = (long)Math.sqrt(2*n);

        long res = n - (x*(x-1))/2;

        res=res%x==0?x:res%x;

        System.out.println(res);

    } }
```

Logical Test Cases

Test Case 1

INPUT (STDIN)

10

EXPECTED OUTPUT

4

Test Case 2

INPUT (STDIN)

99991

EXPECTED OUTPUT

310

62. Question description

After finishing eating her bun, Amanda came up with two integers n and m . She decided to write down two columns of integers — the first column containing integers from 1 to n and the second containing integers from 1 to m . Now the girl wants to count how many pairs of integers she can choose, one from the first column and the other from the second column, such that their sum is divisible by 5.

Formally, Amanda wants to count the number of pairs of integers (x, y) such that $1 \leq x \leq n$, $1 \leq y \leq m$ and $(x + y) \bmod 5$ equals 0.

As usual, Amanda has some troubles and asks you to help.

Constraints:

$1 \leq n, m \leq 1\,000\,000$

Input Format:

The only line of the input contains two integers n and m .

Output Format:

Print the only integer - the number of pairs of integers (x, y) such that $1 \leq x \leq n$, $1 \leq y \leq m$ and $(x + y)$ is divisible by 5.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n,m;

        n = sc.nextInt(); m = sc.nextInt();

        long ans=0;

        for(int i=1;i<=n;i++){

            ans+=m/5;

            if(m%5>=5-(i%5))ans++;        }

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

Logical Test Cases

Test Case 1

INPUT (STDIN)

6 12

EXPECTED OUTPUT

14

Test Case 2

INPUT (STDIN)

21 21

EXPECTED OUTPUT

88

63. Question description

Luke Skywalker got locked up in a rubbish shredder between two presses. R2D2 is already working on his rescue, but Luke needs to stay alive as long as possible. For simplicity we will assume that everything happens on a straight line, the presses are initially at coordinates 0 and L , and they move towards each other with speed v_1 and v_2 , respectively. Luke has width d and is able to choose any position between the presses. Luke dies as soon as the distance between the presses is less than his width. Your task is to determine for how long Luke can stay alive.

Constraints:

$$1 \leq d, L, v_1, v_2 \leq 10000$$

$$d < L$$

Input Format:

The first line of the input contains four integers d, L, v_1, v_2 — Luke's width, the initial position of the second press and the speed of the first and second presses, respectively.

Output Format:

Print a single real value - the maximum period of time Luke can stay alive for.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        long d,L,v1,v2;

        d=sc.nextLong();

        L=sc.nextLong();

        v1=sc.nextLong();

        v2=sc.nextLong();

        System.out.println(String.format("%.4f",(double)(L-d)/(v1+v2)));    }    }
```

Logical Test Cases

Test Case 1

INPUT (STDIN)

1 9 1 2

EXPECTED OUTPUT

2.6667

Test Case 2

INPUT (STDIN)

123 1242 12 312

EXPECTED OUTPUT

3.4537

64. Question description

A soldier wants to buy w apples in the shop. He has to pay k dollars for the first apple, $2k$ dollars for the second one and so on (in other words, he has to pay $i \cdot k$ dollars for the i th apple).

He has n dollars. How many dollars does he have to borrow from his friend soldier to buy w apples?

Constraints:

$$1 \leq k, w \leq 1000$$

$$0 \leq n \leq 10^9$$

Input Format:

The first line contains three positive integers k, n, w , the cost of the first apple, initial number of dollars the soldier has and number of apples he wants.

Output Format:

Output one integer - the amount of dollars that the soldier must borrow from his friend. If he doesn't have to borrow money, output 0.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int k,w,n,a,x=0;

        k=sc.nextInt();

        n=sc.nextInt();

        w=sc.nextInt();

        w=w*(w+1)/2;

        a=k*w-n;

        if(a<=0)

            a=0;    System.out.println(a);    }    }
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1 5 6

EXPECTED OUTPUT

16

Test Case 2

INPUT (STDIN)

20 43 3

EXPECTED OUTPUT

77

65. Question description

Vincent owns a grocery store and suddenly the billing machine malfunctioned.

He has to do the billing manually now and urgently needs a basic calculator

Can you help him by writing a code for a simple calculator which performs arithmetic operations like subtraction, addition, multiplication, division.

Constraints

1sx,y≤1000

Input Format:

Single line input consists of two numbers which are either integer or decimal value, a character input of arithmetic operation as +,-,*,/ in between both of them (45+24=69)

Output Format:

Print the corresponding value of the result of the arithmetic operation

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        float x,y;

        float result;

        char operator;

        x = sc.nextFloat();

        operator = sc.next().charAt(0);

        y = sc.nextFloat();

        switch (operator) {

            case '+':

                result = x+y;

                System.out.println(String.format("%.2f",result));

                break;

            case '-':

                result = x-y;

                System.out.println(String.format("%.2f",result));

                break;
```

```
case '*':  
    result = x*y;  
    System.out.println(String.format("%.2f",result));  
    break;  
case '/':  
    result = x/y;  
    System.out.println(String.format("%.2f",result));  
    break;  
default:  
    System.out.println("Invalid operator!");  
    break;  
}  
}  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
99.7 * 92.6	978.3 / 45
EXPECTED OUTPUT	EXPECTED OUTPUT
9232.22	21.74

66. Question Description:

Brian has a riddle for you and if you manage to solve it, he will give you a cheese burger for free.

You are given an integer n . You need to find two integers l and r such that $l + (l+1) + \dots + (r-1) + r = n$.

Constraints

$1 \leq t \leq 10^4$

$1 \leq n \leq 10^{18}$

$-10^{18} < l < r < 10^{18}$

Input Format:

The first line contains a single integer t the number of test cases.

The first and only line of each test case contains a single integer n

Output Format:

For each test case, print the two integers l and r such that $l + (l+1) + \dots + (r-1) + r = n$.

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args){
        Scanner s= new Scanner(System.in);

        long a,n;

        n = s.nextInt();

        for(int i=0;i<n;i++) {

            a = s.nextLong();
            System.out.println(-(a-1)+" "+a);
        } } }
```

▼ Logical Test Cases

Test Case 1

INPUT (STDIN)

6
1
2
3
3
6
100
25

EXPECTED OUTPUT

0 1
-1 2
-2 3
-5 6
-99 100
-24 25

Test Case 2

INPUT (STDIN)

5
7
2
4
8
90

EXPECTED OUTPUT

-6 7
-1 2
-3 4
-7 8
-89 90

67. Question description:

Attendance is very important for any student to take a semester exam. Bruce is the class monitor and he has been assigned the task to make a program that can display the status. Given the number of classes attended (NA) and a total number of classes conducted (NC). Let P be the percentage of attendance. Its status needs to be printed as below.

Can you help him with the task?

$90 \leq P \leq 100$ = Very Good $80 \leq P \leq 89$ = Good

$75 \leq P \leq 79$ = Average

$P \leq 75$ = Danger

Constraints:

$1 \leq T \leq 100$

$1 \leq NA \leq 1000$

$1 \leq NC \leq 1000$

Input Format:

First line contains number of test cases(T).

Next T lines contain NA and NC separated by a space.

Output Format:

Print the status in a single line for each T.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        float a,c,ans=0;

        int t = input.nextInt();

        while(t-->0)

        {

            a = input.nextFloat();

            c = input.nextFloat();

            ans = (a/c)*100;
```

```

if(ans>=90 && ans<=100){
System.out.println("Very Good");
}

else if(ans>=80 && ans<=89){
System.out.println("Good");
}

else if(ans>=75 && ans<=79){
System.out.println("Average");
}

else if(ans<=75){
System.out.println("Danger");
}

}

}

}

}

```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```

4
520 525
300 400
320 350
400 600

```

EXPECTED OUTPUT

```

Very Good
Average
Very Good
Danger

```

Test Case 2

INPUT (STDIN)

```

3
420 450
220 300
320 350

```

EXPECTED OUTPUT

```

Very Good
Danger
Very Good

```

68. Question description

Marc has been given n numbers, he can perform the following operation any number of times:

- Choose any subset of the numbers (possibly empty), none of which are 0.
- Decrement the numbers in the subset by 1, and increment the numbers not in the subset by K .

Is it possible to perform operations such that exactly $n - 1$ numbers become 0 ?

Please help him out on this because he seems to be confused....

Constraints:

$$2 \leq n \leq 100$$

$$1 \leq K \leq 10$$

$$0 \leq A_i \leq 1000$$

Input Format:

The first line of a test case contains the numbers n and K .

The next line contains n numbers $a_1, a_2, 3, \dots, a_n$

Output Format:

Output "YES" if there is a sequence of operations as described, and "NO" otherwise.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n,a,k=1000,j,ans=0,x;

        int []b=new int[100];

        int []res=new int[k+1];

        n = input.nextInt();

        a = input.nextInt();

        for(j=0;j<k+1;j++)res[j]=0;

        for(j=0;j<n;j++){

            b[j]=input.nextInt();

            x=(b[j]%(a+1));

            res[x]++;

            if(res[x]>n-2)

                ans=1;
```

```
}  
if(ans>0)  
System.out.println("YES");  
else  
System.out.println("NO");  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

2 1
10 10

EXPECTED OUTPUT

YES

Test Case 2

INPUT (STDIN)

3 2
1 2 3

EXPECTED OUTPUT

NO

69. Question description

A King is planning to attack one of its neighbouring Kingdom. The Kingdom that he is going to attack is undefeated till date. So he went to his priest and asked for his help. The priest told him that the only chance he can win the battle is by creating an army that only contain the soldiers having id numbers which are multiple of 7 as 7 is the lucky number for King and his army. As this priest is widely known for his wisdom King chooses to act upon his advice. He calls his army to the ground and ask Cyril to calculate the total number soldiers that can be selected for the battle. Cyril wants you to help him with this.

Constraints:

$$1 \leq N \leq 1000$$

$$1 < A[i] \leq 10^5$$

Input Format:

First line contains an integer n, representing the number of soldiers present on the ground. Second line contains n integers A1,A2,A3.....n that are space separated representing the id number of each soldier.

Output Format:

Output will be an integer giving the total number of soldiers that can go for battle.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a,count=0;

        a = sc.nextInt();

        int[] arr = new int[1000];

        for(int j=0;j<a;j++){

            arr[j]= sc.nextInt();

            if(arr[j]%7==0){

                count++;

            }

        }

        System.out.println(count);

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
6
3 21 19 49 18 28
```

EXPECTED OUTPUT

```
3
```

Test Case 2

INPUT (STDIN)

```
4
12 43 7 2
```

EXPECTED OUTPUT

```
1
```

70. Question description

One of my friend Aidan loves coins and he is playing with them but unable to do one problem. The problem is that you have N coins, not all of different denominations and you have to put the coins of same denomination on top of each other to form pile of different lengths. You have to print the denomination and number of coins in the pile which have maximum number of coins. If 2 or more such denominations are possible then print the denomination having the smallest denomination

Constraints:

$1 \leq N \leq 100000$

Denomination of each coin = x where $1 \leq x \leq 1000$

Input Format:

The first line of the input contains an integer N denoting the number of coins.

The next lines contains N coins.

Output Format:

Print the two numbers as asked in the problem. Print First number on first line and second on second line.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int max=0,element_having_max_freq = 0;

        int n = sc.nextInt();

        int t = n;

        int[] a = new int[100];

        for(int i=0;i<n;i++){
            a[i]=sc.nextInt();
            int count = 0;
            for(int j=0;j<t;j++) {
                if (a[i] == a[j]) {
                    count++;
                }
            }
            if(count>max) {
                max = count;
                element_having_max_freq = a[i];
            }
        }

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

```
System.out.println(max);
```

```
}
```

```
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3  
1 1 2
```

EXPECTED OUTPUT

```
1  
2
```

Test Case 2

INPUT (STDIN)

```
4  
7 7 8 9
```

EXPECTED OUTPUT

```
7  
2
```

71. Question description

Flynn has been given a following process by his classmate as challenge

There is a platform with n columns. 1×1 squares are appearing one after another in some columns on this platform. If there are no squares in the column, a square will occupy the bottom row. Otherwise a square will appear at the top of the highest square of this column.

When all of the n columns have at least one square in them, the bottom row is being removed. You will receive 1 point for this, and all the squares left will fall down one row.

His task is to calculate the amount of points you will receive. Can you help him out?

Constraints:

$1 \leq n, m \leq 1000$

Input Format:

The first line of input contain 2 integer numbers n and m - the length of the platform and the number of the squares.

The next line contain m integer numbers c_1, c_2, \dots, c_m -column in which i -th square will appear.

Output Format:

Print one integer - the amount of points you will receive.

Code:

```
import java.util.*;
```

```
public class Class815427160771 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int numofcolumns,numofsquares;  
        numofcolumns = sc.nextInt();  
        numofsquares = sc.nextInt();  
        int[] columns = new int[numofcolumns];  
        for(int i=0;i<numofsquares;i++) {  
            columns[sc.nextInt()-1]++;  
        }  
        Arrays.sort(columns);  
        System.out.println(columns[0]);  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3 9
1 1 2 2 2 3 1 2 3
```

EXPECTED OUTPUT

```
2
```

Test Case 2

INPUT (STDIN)

```
3 12
1 1 1 1 2 2 2 2 3 3 3 3
```

EXPECTED OUTPUT

```
4
```

72. Question description

Derek and Gavin are busy during the summer, working together to build some ML Models. However, they got stuck in a very complicated problem and are trying to figure out the method. Derek has got two arrays (Let's suppose, we denote them as A and B) of length N, containing single digit numbers only.

Derek needs to combine the i-th element of array -A and B and find the greatest among the two combinations (two digit number formed by joining the digits).

Help Derek!

Constraints:

$1 \leq N \leq 10^5$

Input Format:

The number N-number of digits.

The next two lines contains N space-separated integers (Let's denote it X) - the digits in (0-9) range

Output Format:

print the newly formed N two digit numbers in a space - separated format

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int[] arr1=new int[n];

        int[] arr2=new int[n];

        for(int k=0;k<n;k++) {
            arr1[k]=sc.nextInt();
        }

        for(int k=0;k<n;k++) {
            arr2[k]=sc.nextInt();
        }

        for(int k=0;k<n;k++){

            if(arr1[k]>arr2[k]){
```

```

System.out.print(arr1[k]);
System.out.print(arr2[k]+" ");
}

else if(arr1[k]==arr2[k]){
System.out.print(arr1[k]);
System.out.print(arr2[k]+" ");
}

else if(arr1[k]<arr2[k]){
System.out.print(arr2[k]);
System.out.print(arr1[k]+" ");
}

}

}

}

}

```

Logical Test Cases

Test Case 1

INPUT (STDIN)

```

5
1 2 3 4 5
3 2 1 4 5

```

EXPECTED OUTPUT

```

31 22 31 44 55

```

Test Case 2

INPUT (STDIN)

```

6
9 0 1 2 3 6
4 4 5 3 2 1

```

EXPECTED OUTPUT

```

94 40 51 32 32 61

```


73. Question description

Robert and Winston decided to compete in text typing on the site "Keyboard Racing". During the competition, they have to type a text consisting of s characters. The first participant types one character in v_1 milliseconds and has ping p_1 milliseconds. The second participant types one character in v_2 milliseconds and has ping p_2 milliseconds.

If connection ping (delay) is p milliseconds, the competition passes for a participant as follows:

1. Exactly after p milliseconds after the start of the competition the participant receives the text to be entered.
2. Right after that he starts to type it.
3. Exactly t milliseconds after he ends typing all the text, the site receives information about it.

The winner is the participant whose information on the success comes earlier. If the information comes from both participants at the same time, it is considered that there is a draw.

Given the length of the text and the information about participants, determine the result of the game.

Constraints:

$$1 \leq s, v_1, v_2, p_1, p_2 \leq 1000$$

Input Format:

The first line contains five integers s, v_1, v_2, p_1, p_2 . the number of characters in the text, the time of typing one character for the first participant, the time of typing one character for the the second participant, the ping of the first participant and the ping of the second participant.

Output Format:

If the Robert wins, print "Robert". If the Winston participant wins, print "Winston". In case of a draw print "Friendship".

Code:

```
import java.util.*;

class KeyboardRace{
    int numofchar,time1,time2,ping1,ping2;
}

public class Class815427160771{

    public static void main(String args[]){
        KeyboardRace participant=new KeyboardRace();

        Scanner sc = new Scanner(System.in);
```

```

participant.numofchar= sc.nextInt();
participant.time1= sc.nextInt();
participant.time2= sc.nextInt();
participant.ping1= sc.nextInt();
participant.ping2= sc.nextInt();

if((participant.time1*participant.numofchar)+participant.ping1+participant.ping1<(participant.time2
*participant.numofchar)+participant.ping2+participant.ping2)

System.out.print("Robert");

else if((participant.time1*participant.numofchar) + participant.ping1+participant.ping1 >
(participant.time2*participant.numofchar)+participant.ping2+participant.ping2)

System.out.print("Winston");

else

System.out.print("Friendship");
}
}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
5 1 2 1 2	8 8 4 2 6
EXPECTED OUTPUT	EXPECTED OUTPUT
Robert	Winston

74. Question description

On vacations n pupils decided to go on excursion and gather all together. They need to overcome the path with the length l /meters. Each of the pupils will go with the speed equal to v_1 . To get to the excursion quickly, it was decided to rent a bus, which has seats for k people (it means that it can't fit more than k people at the same time) and the speed equal to v_2 . In order to avoid seasick, each of the pupils want to get into the bus no more than

once.

Determine the minimum time required for all n pupils to reach the place of excursion. Consider that the embarkation and disembarkation of passengers, as well as the reversal of the bus, take place immediately and this time can

be neglected.

Constraints:

$$1 \leq n \leq 10000$$

15/109

Input Format:

The first line of the input contains five positive integers n , l , v_1 , v_2 and k - the number of pupils, the distance from meeting to the place of excursion, the speed of each pupil, the speed of bus and the number of seats in the bus.

Output Format:

Print the real number - the minimum time in which all pupils can reach the place of excursion.

Code:

```
import java.util.*;

class Excursion{
    long n,k;
}

class Speed{
    double l,v1,v2;
}

public class Class815427160771 {
    public static void main(String[] args) {
        Excursion obj=new Excursion();
        Speed obj1=new Speed();
        Scanner sc = new Scanner(System.in);
        obj.n= sc.nextLong();
```

```

obj1.l= sc.nextDouble();
obj1.v1= sc.nextDouble();
obj1.v2= sc.nextDouble();
obj.k= sc.nextLong();
double m=(obj.n-1)/obj.k+1;
    double v=obj1.v1+obj1.v2;
    double x=obj1.l*v*obj1.v2/(m*v*obj1.v2-(m-1)*(obj1.v2-obj1.v1)*obj1.v2);
    System.out.println(String.format("%.4f",(float)x/obj1.v2+(obj1.l-x)/obj1.v1));
}
}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
5 10 1 2 5	3 6 1 2 1
EXPECTED OUTPUT	EXPECTED OUTPUT
5.0000	4.7143

75. Question description

Henry once received a gift: it was a book about aliens, which he started read immediately. This book had c pages.

At first day Henry read v_0 pages, but after that he started to speed up. Every day, starting from the second, he read a pages more than on the previous day (at first day he read v_0 pages, at second — $v_0 + a$ pages, at third — $v_0 + 2a$ pages, and so on). But Henry is just a human, so he physically wasn't able to read more than v_i pages per day.

Also, to refresh his memory, every day, starting from the second, Henry had to reread last r pages he read on the previous day. Henry finished the book when he read the last page for the first time.

Help Henry to calculate how many days he needed to finish the book.

Constraints:

$$1 \leq c \leq 1000$$

$$0 \leq v_0 < v_i \leq 1000$$

$$0 \leq a \leq 1000$$

Input Format:

First and only line contains five space-separated integers: c , v_0 , v_i , a and r - the length of the book in pages, the initial reading speed, the maximum reading speed, the acceleration in reading speed and the number of pages for rereading.

Output Format:

Print one integer - the number of days Henry needed to finish the book.

Code:

```
import java.util.*;

class ReadingBooks{
    int length,speed,maxspeed,acc,reread;
}

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        ReadingBooks book=new ReadingBooks();
        book.length= sc.nextInt();
        book.speed=sc.nextInt();
        book.maxspeed= sc.nextInt();
        book.acc=sc.nextInt();
        book.reread=sc.nextInt();

        int ans=1,cur=book.speed,k=1,vel=book.speed;
        while(cur<book.length){
```

```
ans++;  
vel=Math.min(book.maxspeed,vel+book.acc);  
cur+=(vel-book.reread);  
k++;  
}  
System.out.println(ans);  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

12 4 12 4 1

EXPECTED OUTPUT

3

Test Case 2

INPUT (STDIN)

5 5 10 5 4

EXPECTED OUTPUT

1

76. Question description

To make a paper boat, one has to use a rectangular piece of paper. From a sheet of standard size you can make s boats.

A group of k people decided to make n boats each. They are going to buy several packs of paper, each of them containing p sheets, and then distribute the sheets between the people. Each person should have enough sheets to make n boats. How many packs should they buy?

Constraints:

$$1 \leq k, n, s, p \leq 109$$

Input Format:

The only line contains four integers k, n, s, p - the number of people, the number of boats each should make, the number of boats that can be made using one sheet and the number of sheets in one pack, respectively.

Output Format:

Print a single integer - the minimum number of packs they should buy.

Code:

```
import java.util.Scanner;

class PaperBoat{
    int k,n,s,p; }

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        PaperBoat fun=new PaperBoat();
        fun.k= sc.nextInt(); fun.n= sc.nextInt(); fun.s= sc.nextInt(); fun.p= sc.nextInt();
        int perPerson=(int) Math.ceil(fun.n/(double)fun.s);
        System.out.println((int)Math.ceil((fun.k*perPerson)/(double)fun.p)); } }
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

5 3 2 3

EXPECTED OUTPUT

4

Test Case 2

INPUT (STDIN)

5 3 100 1

EXPECTED OUTPUT

5

77. Question description

During the winter holidays, the demand for Christmas balls is exceptionally high. The advances in alchemy allow easy and efficient ball creation by utilizing magic crystals.

Hailey needs to obtain some yellow, green and blue balls. It's known that to produce a yellow ball one needs two yellow crystals, green — one yellow and one blue, and for a blue ball, three blue crystals are enough.

Right now there are A yellow and B blue crystals in Hailey's disposal. Find out how many additional crystals he should acquire in order to produce the required number of balls.

Constraints:

$0 \leq A, B \leq 10^9$

$0 \leq x, y, z \leq 10^9$

Input Format:

The first line features two integers A and B, denoting the number of yellow and blue crystals respectively at Hailey's disposal.

The next line contains three integers x, y and z- the respective amounts of yellow, green and blue balls to be obtained.

Output Format:

Print a single integer - the minimum number of crystals that Hailey should acquire in addition.

Code:

```
import java.util.Scanner;

class ChristmasBalls{
    long A,B,x,y,z;
}

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ChristmasBalls happy=new ChristmasBalls();
        happy.A= sc.nextLong();
        happy.B= sc.nextLong();
        happy.x= sc.nextLong();
        happy.y= sc.nextLong();
        happy.z= sc.nextLong();
        long ans=0;
```



```
if(2*happy.x+happy.y>happy.A) {  
    ans+=((2*happy.x)+happy.y)-happy.A;  
}  
if(3*happy.z+happy.y>happy.B) {  
    ans+=((3*happy.z+happy.y)-happy.B);  
}  
System.out.println(ans);  
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
4 3  
2 1 1
```

EXPECTED OUTPUT

```
2
```

Test Case 2

INPUT (STDIN)

```
100 10  
2 3 4
```

EXPECTED OUTPUT

```
5
```

78. Problem Description:

Krishna has just arrived in the city of Madhura.

He brought an old house and renovating it. On seeing the pathetic floor conditions he planned to pave it with tile.

He has a $m \times n$ units of floor area and want to cover it up with 2×1 size tiles.

Krishna is no so good at calculations.

Could you help him to find out the minimum number tiles he needs to cover the floor?

Constraints:

$1 \leq m, n \leq 500$

Input Format:

Only line of input has two integers m and n separated by as space.

Output Format:

Print the minimum number of tiles need to pave the floor as output.

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int m = input.nextInt();
        int n = input.nextInt();

        float m1 = (float)m;
        float n1 = (float)n;

        System.out.println(String.format("%.0f", (m1*n1/2)));}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

121 175

EXPECTED OUTPUT

10588

Test Case 2

INPUT (STDIN)

183 156

EXPECTED OUTPUT

14274

79. Problem Description:

Tina successfully completed her first Month as Navy Officer and her much awaited first month salary got credited into her account and also got her salary report in her mail.

On seeing the salary report there is no information available about the Employee provident fund. So Tina would like to know about the Employee and Employer Contribution for EPF. Can you help her?

Constraints:

$1 \leq \text{basicPay} \leq 50000$

$1 \leq \text{employeeFund} \leq 17.5$

$1 \leq \text{employerFund} \leq 23.5$

Input Format:

The only line of input has a single floating point value representing the basic salary of Tina.

Output Format:

In the First Line of output print the Employee Share of EPF with only two values after decimal point.

In the Second Line of output print the Employer Share of EPF with only two values after decimal point.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double employeeFund, employerFund;

        double basicPay = input.nextDouble();
        employeeFund = (basicPay/100)*17.5;
        employerFund = (basicPay/100)*23.5;
        System.out.println(String.format("%.2f", employeeFund));
        System.out.println(String.format("%.2f", employerFund)); } }
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
43982.56	29717.19
EXPECTED OUTPUT	EXPECTED OUTPUT
7696.95 10335.90	5200.51 6983.54

80. Problem Description:

Aarav and Nathan who live in the fictional town of Pune, think and do innovatively on weekends. Every day the boys embark on some grand new project, which annoys their controlling sister Thendral, who tries to bust them.

One Sunday they were both sitting under a tree in their back yard. They decide to invent the machine which would allow us to enter 2 numbers it would say whether one of the entered numbers is an appropriate value of the other number entered.

They decide to insert a program code into the machine. A number is said to be an approximate value of the other if they differ by utmost 0.5.

Can you help Aarav and Nathan with the programming logic to find whether the given number is an approximate number of other?

Constraints:

$0 \leq A \leq 1000$

$0 < B \leq 1000$

Input Format:

The only line of input has two numbers of type float separated by a space representing A and B respectively.

Output Format:

In the only line of output print as "Approximate number" or "Not an Approximate number" based on the condition.

Code:

```
import java.util.Scanner;

public class Class815427160771 { public static void main(String[] args) {

Scanner input = new Scanner(System.in);

float a = input.nextFloat(); float b = input.nextFloat();

if((int) a== (int)b) System.out.println("Approximate number");

else System.out.println("Not an Approximate number"); }}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

15.1
15.6

EXPECTED OUTPUT

Approximate number

Test Case 2

INPUT (STDIN)

33.5
31.3

EXPECTED OUTPUT

Not an Approximate number

81. Problem Description:

Zaheer and Vinod went to California for attending the Apple products launch event.

In the event location there are lot of mini contests going on with exciting rewards.

In one such stall there was a contest where they have given a floating point number and asked the visitors to display the rightmost digit integral part of the number using a programming logic.

Can you help Zaheer and Vinod to display the rightmost integral part of the number so that they get the exciting apple products as reward?

Constraints:

$1.000 < \text{number} < 1000.999$

Input Format:

Only line of input has a single floating point value with three values after decimal point.

Output Format:

Print the output in a single line rightmost integral part of the number.

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        float appleno = input.nextFloat();
        System.out.println((int)appleno%10); }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1985.674

EXPECTED OUTPUT

5

Test Case 2

INPUT (STDIN)

3489.781

EXPECTED OUTPUT

9

82. Problem Description:

Sathya is an mathematical expert training youngsters struggling in maths to make them better.

Sathya usually gives interesting problems to the youngsters to make them love the math.

One such day Sathya provided the youngsters the base number and exponent and asked the power of that given number.

Youngsters were lacking the idea to solve the problem.

Being a exciting youngster can you solve it?

Constraints:

$1.0 < \text{base} < 20.0$

$1.0 \leq \text{exp} \leq 10.0$

Input Format:

The first line of the input is a base number

The second line of the input is an exponent.

Output Format:

Print the power of a number as output in a single line.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double base = input.nextDouble();

        double exp = input.nextDouble();

        System.out.println(String.format("%.2f",Math.pow(base, exp))); } }
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
7.5 6.3	17.2 2.6
EXPECTED OUTPUT	EXPECTED OUTPUT
325751.15	1630.70

83. Problem Description:

Arav was a popular maths trainer, he gave a 4 digit number to his students as an assignment.

The Students has to identify ones portion of given number.

But students are confused with the logic for doing so.

Can you help the students with the appropriate logic?

Constraint:

$1000 \leq \text{num} \leq 2600$

Input Format:

Only line of input has a single integer representing "num";

Output Format:

Print the Digit at one's place

Explanation:

Let us say Aarav given a number "7821" then the number at one place is "1"

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int num = input.nextInt();

        System.out.println(num%10); } }
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

8524

EXPECTED OUTPUT

4

Test Case 2

INPUT (STDIN)

7425

EXPECTED OUTPUT

5

84. Problem Description:

Nancy is a data scientist. She regularly faces about Terra bytes of data in her work.

One day she was working on an application that collects users address and stores it based on the type of field it has to be .

Unfortunately the application malfunctioned and the data collapsed .

Nancy now has the burden of arranging the users data into their respective field can you help her?

Constraint:

$0000 < \text{hno} \leq 9999$

$100000 < \text{pincode} \leq 999999$

$1000 \leq \text{employeeID} \leq 9999$

$000 \leq \text{areacode} \leq 999$

Input Format:

First line of input represents hno

Second line of input represents pincode

Third line of input represents employeeID

Fourth line of input represents areacode

Output Format:

Print the output as per the format specification in the testcases.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int hno = input.nextInt();

        int pincode = input.nextInt();

        int employeeID = input.nextInt();

        int areacode = input.nextInt();

        System.out.println("EmployeeID : "+employeeID);

        System.out.println("Area Code : "+areacode);

        System.out.println("House Number : "+hno);
```



```
System.out.println("Pincode : "+pincode);
```

```
}
```

```
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
73 613004 103245 14
```

EXPECTED OUTPUT

```
EmployeeID : 103245  
Area Code : 14  
House Number : 73  
Pincode : 613004
```

Test Case 2

INPUT (STDIN)

```
139 613012 109651 19
```

EXPECTED OUTPUT

```
EmployeeID : 109651  
Area Code : 19  
House Number : 139  
Pincode : 613012
```

85. Problem Description:

Binita always dreamed of flying in the sky from her childhood. Her goal was to become a pilot. When she applied for the pilot training program she cleared the entrance exam but failed the physical test due to overweight. She forgot to take care of her health during the preparation. But she had the spirit of not giving up. Binita joined a gym to pass the physical test next time. Her gym trainer asked her to calculate BMI for giving her a proper diet plan, But she had other works to do can you help this motivated girl?

Functional Description:

$BMI = \text{weight} / \text{height} \times \text{height}$

Constraints:

$40 \leq \text{weight} \leq 80$

$1.50 < \text{Height} < 1.72$

Input Format:

First line has the value of weight of type integer

Second line has the value of height of type float

Output Format:

Print the BMI with only two values after decimal points.

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int weight = input.nextInt();
        float height = input.nextFloat();
        System.out.println(String.format("%.2f", weight / (height * height))); } }
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
64 1.69	89 1.59
EXPECTED OUTPUT	EXPECTED OUTPUT
22.41	35.20

86. Problem Description:

Shiva is part of the popular construction company in Tamilnadu.

They constructed an apartment on the express highway.

The apartment is Trapezium in size.

Shiva is part of budget estimation team so he would like to calculate the Area of that apartment.

Can you help him?

Constraints:

$1 < \text{base1} < 500$

$1 < \text{base2} < 500$

$1 < \text{height} < 500$

Input Format:

Only Line of input has three floating point values representing base1 base2 and height separated by a space

Output Format:

Print the area of the apartment in a single line with two values after decimal point.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double base1 = input.nextDouble();

        double base2 = input.nextDouble();

        double height = input.nextDouble();

        System.out.println(String.format("%.2f", (base1+base2)*height/2)); } }
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

100.12 126.76 300.12

EXPECTED OUTPUT

34045.61

Test Case 2

INPUT (STDIN)

89.63 62.54 78.28

EXPECTED OUTPUT

5955.93

87. Problem Description

Nedumaran Rajangam is an visionary in Aviation industry. His wife Bommi runs a bakery where she sells loaves of bread for 185 rupees each.

Bommi provided the discount of 60 % for a Day old bread.

Bommi noted the number of loaves of day old bread being purchased from the customer.

Bommi likes to know the following details regarding the sales in her bakery:

The income if n number of loaves sold on its regular

The amount that will be discounted for customers if discount of 60% is applied to the regular income

The profit she will get after giving the discount.

Constraints:

BreadPrice = 185

$25 < \text{loaves} \leq 150$

Input format:

Only line of input has a single integer representing the number of the loaves of the bread

Output format:

First Line : Print the Regular Price for n number of loaves.

Second Line: Print the Amount Discounted if 60% discount is applied with 2 values after decimal point.

Third Line: Print the amount to be paid after discount with 2 values after decimal point.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int loaves = input.nextInt();

        int regular = loaves*185;

        double discount = regular*(0.6);

        double total = regular - discount;

        System.out.println("Regular Price="+regular);

        System.out.println("Amount Discounted="+ String.format("%.2f",discount));

        System.out.println("Amount to be paid="+ String.format("%.2f",total));

    }

}
```

}

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

25

EXPECTED OUTPUT

```
Regular Price=4625
Amount
Discounted=2775.00
Amount to be
paid=1850.00
```

Test Case 2

INPUT (STDIN)

56

EXPECTED OUTPUT

```
Regular Price=10360
Amount
Discounted=6216.00
Amount to be
paid=4144.00
```

88. Question description

Imp likes his plush toy a lot.



Recently, he found a machine that can clone plush toys. Imp knows that if he applies the machine to an original toy, he additionally gets one more original toy and one copy, and if he applies the machine to a copied toy, he gets two additional copies.

Initially, Imp has only one original toy. He wants to know if it is possible to use machine to get exactly x copied toys and y original toys? He can't throw toys away, and he can't apply the machine to a copy if he doesn't currently have any copies.

Constraints:

$0 \leq x, y \leq 10^9$

Input Format:

The only line contains two integers x and y — the number of copies and the number of original toys Imp wants to get (including the initial one).

Output Format:

Print "Yes", if the desired configuration is possible, and "No" otherwise.

You can print each letter in arbitrary case (upper or lower).

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        int b,h;

        b=sc.nextInt();

        h=sc.nextInt();

        if(h==0 || (h==1 && b>0) || ((h-1)%2!=b%2) || (b<h-1))

            System.out.println("No");

        else

            System.out.println("Yes");

    }

}
```

```
        sc.close();  
    }  
}
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
6 3	4 2
EXPECTED OUTPUT	EXPECTED OUTPUT
Yes	No

89. Question description

It's one more school day now. Sasha doesn't like classes and is always bored at them. So, each day he invents some game and plays in it alone or with friends.

Today he invented one simple game to play with Lena, with whom he shares a desk. The rules are simple. Sasha draws n sticks in a row. After that the players take turns crossing out exactly k sticks from left or right in each turn. Sasha moves first, because he is the inventor of the game. If there are less than k sticks on the paper before some turn, the game ends. Sasha wins if he makes strictly more moves than Lena. Sasha wants to know the result of the game before playing, you are to help him.

Constraints:

$1 \leq n, k \leq 1018$

$k \leq n$

Input Format:

The first line contains two integers n and k - the number of sticks drawn by Sasha and the number k - the number of sticks to be crossed out on each turn.

Output Format:

If Sasha wins, print "YES" (without quotes), otherwise print "NO" (without quotes).

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        long n,k;

        n=sc.nextLong(); k=sc.nextLong();

        String c = n/k%2==1?"YES":"NO";

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

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
6 4	10 4
EXPECTED OUTPUT	EXPECTED OUTPUT
YES	NO

90. Question description

Recently Shankar learned about a special kind of numbers that she calls beautiful numbers. The number is called beautiful iff its binary representation consists of $k + 1$ consecutive ones, and then k consecutive zeroes.

Some examples of beautiful numbers:

12 (110);

1102 (610);

- 11110002 (12010);
- 1111100002 (49610).

More formally, the number is beautiful iff there exists some positive integer k such that the number is equal to $(2-1) * (2k-1)$.

Shankar has got an integer number n , and she wants to find its greatest beautiful divisor. Help her to find it!

Constraints:

$1 \leq n \leq 10^5$

Input Format:

The only line of input contains one number n - number Shankar has got.
the

Output Format:

Output one number - the greatest beautiful divisor of Shankar's number. It is obvious that the answer always exists.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n,r=0;

        n = input.nextInt();

        for(int k=1;k<10;k++){

            int x=(int) ((Math.pow(2,k)-1)*Math.pow(2,k-1));

            if(n%x==0){

                r=x;

            }

        }

    }

}
```

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

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

992

EXPECTED OUTPUT

496

Test Case 2

INPUT (STDIN)

76920

EXPECTED OUTPUT

120

91. Question description

It seems that Dana is seriously sick. He is going visit n doctors to find out the exact diagnosis. Each of the doctors needs the information about all previous visits, so Dana has to visit them in the prescribed order (i.e. Dana should first visit doctor 1, then doctor 2, then doctor 3 and so on). Dana will get the information about his health from the last doctor.

Doctors have a strange working schedule. The doctor i goes to work on the s_i day and works every d_i day. So, he works on days $s_i, s_i + d_i, s_i + 2d_i, \dots$

The doctor's appointment takes quite a long time, so Dana can not see more than one doctor per day. What is the minimum time he needs to visit all doctors?

Constraints:

$1 \leq n \leq 1000$

$1 \leq s_i, d_i \leq 1000$

Input Format:

First line contains an integer n - number of doctors.

Next n lines contain two numbers s_i and d_i .

Output Format:

Output a single integer — the minimum day at which - Dana can visit the last doctor.

Code:

```
import java.util.Scanner;

public class Class815427160771 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int nDoctors;

        int nDay=0;

        nDoctors = sc.nextInt();

        for (int i = 0; i < nDoctors; i++) {
            int startWork = sc.nextInt();

            int days = sc.nextInt();

            while (startWork <= nDay)
                startWork += days;

            nDay = startWork;
        }

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

```
}  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3  
2 2  
1 2  
2 2
```

EXPECTED OUTPUT

```
4
```

Test Case 2

INPUT (STDIN)

```
2  
10 1  
6 5
```

EXPECTED OUTPUT

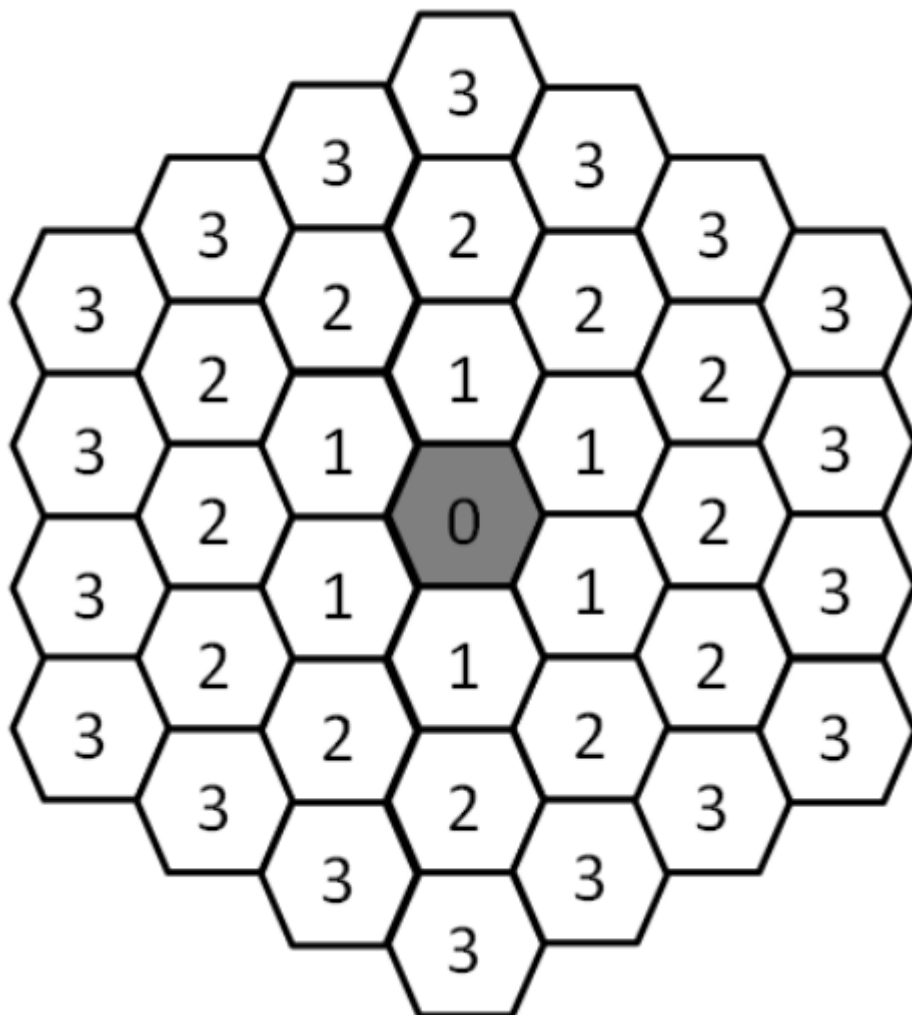
```
11
```

92. Question description

After a probationary period in the game development company of IT City Tony was included in a group of the programmers that develops a new turn-based strategy game resembling the well known "Heroes of Might & Magic". A part of the game is turn-based fights of big squadrons of enemies on infinite fields where every cell is in form of a hexagon.

Some of magic effects are able to affect several field cells at once, cells that are situated not farther than n cells away from the cell in which the effect was applied. The distance between cells is the minimum number of cell border crosses on a path from one cell to another.

It is easy to see that the number of cells affected by a magic effect grows rapidly when n increases, so it can adversely affect the game performance. That's why Tony decided to write a program that can, given n , determine the number of cells that should be repainted after effect application, so that game designers can balance scale of the effects and the game performance. Help him to do it. Find the number of hexagons situated not farther than n cells away from a given cell.



Constraints:

$0 \leq n \leq 109$

Input Format:

The only line of the input contains one integer n.

Output Format:

Output one integer - the number of hexagons situated not farther than n cells away from a given cell.

Code:

```
import java.util.*;

public class Class815427160771{

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        long n=sc.nextLong();

        System.out.println(1+(3*n*n)+(3*n));

    }

}
```

✓ Logical Test Cases

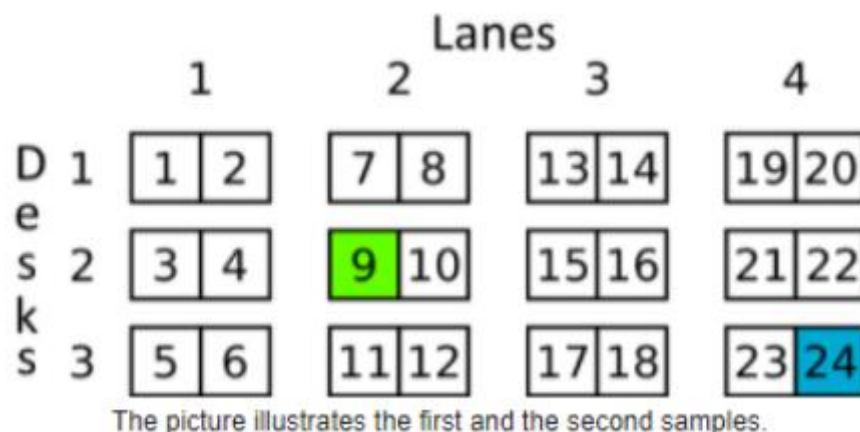
Test Case 1
INPUT (STDIN)
2
EXPECTED OUTPUT
19

Test Case 2
INPUT (STDIN)
3
EXPECTED OUTPUT
37

93. Question description

Santa Claus is the first who came to the Christmas Olympiad, and he is going to be the first to take his place at a desk! In the classroom there are n lanes of m desks each, and there are two working places at each of the desks. The lanes are numbered from 1 to n from the left to the right, the desks in a lane are numbered from 1 to m starting from the blackboard. Note that the lanes go perpendicularly to the blackboard, not along it (see picture).

The organizers numbered all the working places from 1 to $2nm$. The places are numbered by lanes (i. e. all the places of the first lane go first, then all the places of the second lane, and so on), in a lane the places are numbered starting from the nearest to the blackboard (i. e. from the first desk in the lane), at each desk, the place on the left is numbered before the place on the right.



Santa Clause knows that his place has number k . Help him to determine at which lane at which desk he should sit, and whether his place is on the left or on the right!

Constraints:

$$1 \leq n, m \leq 10000$$

$$1 \leq k \leq 2nm$$

Input Format:

The only line contains three integers n , m and k - the number of lanes, the number of desks in each lane and the number of Santa Claus' place.

Output Format:

Print two integers: the number of lane l , the number of desk d , and a character, which stands for the side of the desk Santa Claus. The character s should be "L", if Santa Clause should sit on the left, and "R" if his place is on the right.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
```

```

        int n,m,k;
        int p,l,d;
        n = sc.nextInt();
        m = sc.nextInt();
        k = sc.nextInt();
        p = (int)Math.floor((k-1)/2);
        l = (int)Math.floor(p/m)+1;
        d = (p%m)+1;
        System.out.println(l+" "+d+" "+((k%2==0) ? "R" : "L"));
        sc.close();
    }
}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
4 3 9	4 3 24
EXPECTED OUTPUT	EXPECTED OUTPUT
2 2 L	4 3 R

94. Question description

Aaron and Eric have played several table tennis sets. Each set consists of several serves, each serve is won by one of the players, he receives one point and the loser receives nothing. Once one of the players scores exactly k points, the score is reset and a new set begins.

Across all the sets Aaron scored a points in total, and Eric scored b points. Given this information, determine the maximum number of sets they could have played, or that the situation is impossible.

Note that the game consisted of several complete sets.

Constraints:

$$1 \leq k \leq 109$$

$$0 < a, b < 10^9$$

$$a + b > 0$$

Input Format:

The first line contains three space-separated integers k , a and b .

Output Format:

If the situation is impossible, print a single number "Impossible". Otherwise, print the maximum possible number of sets.

Code:

```
import java.util.*;

public class Class815427160771{

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        int k,a,b;

        k = sc.nextInt();

        a = sc.nextInt();

        b = sc.nextInt();

        if(a%k!=0&&k>=b || b<k) System.out.println("Impossible");

        else if(b%k!=0&&a<k) System.out.println("Impossible");

        else{

            int ans = a/k + b/k;

            System.out.println(ans);

        }

    }

}
```

}

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1 5 9

EXPECTED OUTPUT

14

Test Case 2

INPUT (STDIN)

11 2 3

EXPECTED OUTPUT

Impossible

95. Question description

You are asked to watch your nephew who likes to play with toy blocks in a strange way. He has n boxes and the i -th box has a_i blocks. His game consists of two steps:

1. he chooses an arbitrary box i ;
2. he tries to move all blocks from the i -th box to other boxes.

If he can make the same number of blocks in each of $n-1$ other boxes then he will be happy, otherwise, will be sad. Note that your nephew can only move the blocks from the chosen box to the other boxes; he cannot move blocks from the other boxes.

You don't want to make your nephew sad, so you decided to put several extra blocks into some boxes in such a way that no matter which box i he chooses he won't be sad. What is the minimum number of extra blocks you need to put?

Constraint:

$$1 \leq a_i \leq 1000$$

$$2 \leq n \leq 105$$

$$\sum a_i \leq 10^9$$

Input Format:

The first line of each test case contains the integer n - the number of boxes.

The second line of each test case contains n integers a_1, a_2, \dots, a_n - the number of blocks in each box.

It's guaranteed that the sum of n over test cases doesn't exceed 105.

Output Format:

For each test case, print a single integer - the minimum number of blocks you need to put. It can be proved that the answer always exists, i. e. the number of blocks is finite.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        long s,m,now;

        int n = sc.nextInt();

        int arr[] = new int[n];

        s = 0;

        m = 0;

        for (int i=0;i<n;i++){
```

```

        arr[i] = sc.nextInt();
        s+=arr[i];
        m = Math.max(arr[i],m);
    }
    now = Math.max(m*(n-1), ((long)Math.ceil((s*1.0)/(n-1)))*(n-1));
    System.out.println(now-s);
}
}

```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
3 3 2 2	3 0 3 0
EXPECTED OUTPUT	EXPECTED OUTPUT
1	3

96. Question description

Your favorite grocery store sells n Zinder Surprise chocolate eggs. You know that exactly s stickers and exactly t toys are placed in n eggs in total.

Each Zinder Surprise can be one of three types:

- it can contain a single sticker and no toy; ● it can contain a single toy and no sticker;
- it can contain both a single sticker and a single toy.

But you don't know which type a particular Zinder Surprise has. All eggs look identical and indistinguishable from each other.

What is the minimum number of Zinder Surprise Eggs you have to buy to be sure that, whichever types they are, you'll obtain at least one sticker and at least one toy?

Note that you do not open the eggs in the purchasing process, that is, you just buy some number of eggs.

It's guaranteed that the answer always exists.

Constraints:

$$1 \leq n \leq 10^9$$

$$1 \leq s, t \leq n$$

$$s + t \leq 2n$$

Input Format:

Single line contain three integers n , s and t - the number of eggs, stickers and toys.

All queries are independent.

Output Format:

Print the minimum number of Zinder Surprise Eggs you have to buy to be sure that, whichever types they are, you'll obtain at least one sticker and one toy

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        int x,y,z;

        int c;

        x = sc.nextInt();

        y = sc.nextInt();

        z = sc.nextInt();
```

```
        if (y <= z)
        {
            c = (x - y) + 1;
        }
        else
        {
            c = (x - z) + 1;
        }
        System.out.println(c);
    }
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

10 5 7

EXPECTED OUTPUT

6

Test Case 2

INPUT (STDIN)

10 10 10

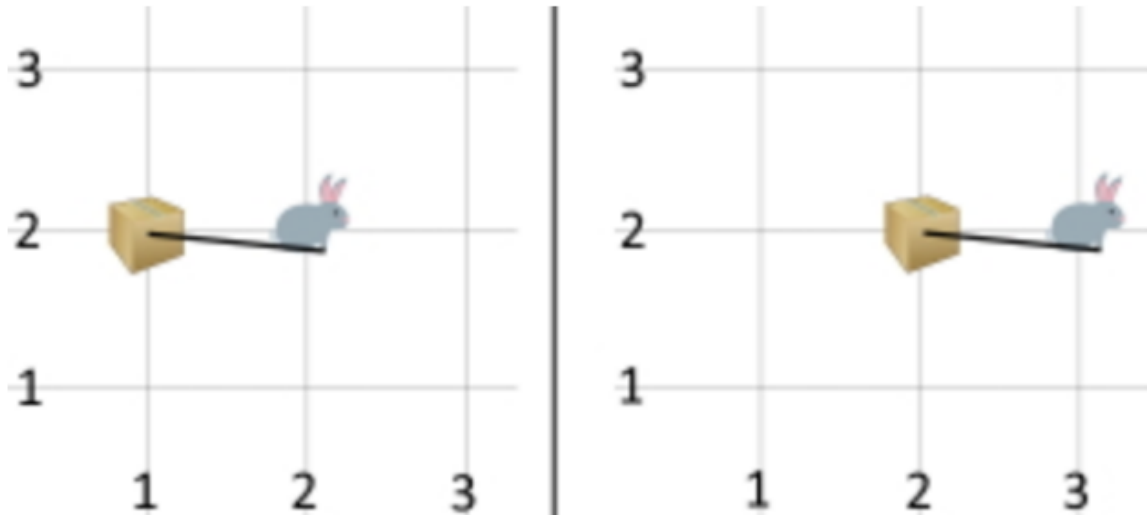
EXPECTED OUTPUT

1

97. Question description

B Rabbit is trying to move a box containing food for the rest of the zoo in the coordinate plane from the point (x_1, y_1) to the point (x_2, y_2) .

He has a rope, which he can use to pull the box. He can only pull the box if he stands exactly 1 unit away from the box in the direction of one of two coordinate axes. He will pull the box to where he is standing before moving out of the way in the same direction by 1 unit.



For example, if the box is at the point $(1, 2)$ and Wabbit is standing at the point $(2, 2)$, he can pull the box right by 1 unit, with the box ending up at the point $(2, 2)$ and Wabbit ending at the point $(3, 2)$.

Also, Wabbit can move 1 unit to the right, left, up, or down without pulling the box. In this case, it is not necessary for him to be in exactly 1 unit away from the box. If he wants to pull the box again, he must return to a point next to the box. Also, Wabbit can't move to the point where the box is located.

Wabbit can start at any point. It takes 1 second to travel 1 unit right, left, up, or down, regardless of whether he pulls the box while moving.

Determine the minimum amount of time he needs to move the box from (x_1, y_1) to (x_2, y_2) . Note that the point where Wabbit ends up at does not matter.

Constraints:

$1 \leq t \leq 1000$

$1 \leq x_1, y_1, x_2, y_2 \leq 10^9$

Input Format:

Single line contains four space-separated integers x_1, y_1, x_2, y_2 , describing the next test case.

Output Format:

Print a single integer: the minimum time in seconds Wabbit needs to bring the box from (x_1, y_1) to (x_2, y_2) .

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String arg[]) {

        Scanner sc=new Scanner(System.in);

        int x,y,x2,y2;

                x=sc.nextInt();

                y=sc.nextInt();

                x2=sc.nextInt();

                y2=sc.nextInt();

                if(x==x2 || y==y2)

                        System.out.println(Math.abs(x-x2)+Math.abs(y-y2));

                else

                        System.out.println(Math.abs(x-x2)+Math.abs(y-y2)+2);

    }

}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

1 2 2 2

EXPECTED OUTPUT

1

Test Case 2

INPUT (STDIN)

1 1 2 2

EXPECTED OUTPUT

4

98. Question description

Beatrice has been given q queries in the following form:

Given three integers l_i , r_i and d_i , find minimum positive integer x_i such that it is divisible by d_i and it does not belong to the segment $[l_i, r_i]$.

Can you answer all the queries and help her out?

Recall that a number x belongs to segment $[l, r]$ if $l \leq x \leq r$.

Constraints:

$1 \leq q \leq 500$

$1 \leq l_i \leq r_i \leq 10^9$

$1 < d_i \leq 10^9$

Input Format:

The first line contains one integer q - the number of queries.

Then q lines follow, each containing a query given in the format $l_i \ r_i \ d_i$ which are integers.

Output Format:

For each query print one integer: the answer to this query.

Code:

```
import java.util.*;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        int l,r,d;

        int t = s.nextInt();

        while(t>0) {

            l = s.nextInt();

            r = s.nextInt();

            d = s.nextInt();

            if(d<l)

                System.out.println(d);

            else

                System.out.println((r/d)*d+d);

            t--;

        }

    }

}
```

```
}  
    }  
}
```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3  
2 4 2  
5 10 4  
3 10 1
```

EXPECTED OUTPUT

```
6  
4  
1
```

Test Case 2

INPUT (STDIN)

```
2  
1 2 3  
4 6 5
```

EXPECTED OUTPUT

```
3  
10
```

99. Question description

Alexis and Fred play the classic game of odds and evens. In this game, each of the two players choose a number between 1 and 5 without revealing to their opponent. Both of the players then simultaneously reveal their number by holding up that many fingers of their hand. Then the winner is decided based upon whether the sum of numbers played by both the players is odd or even. In this game we assume that if the sum is odd then Alexis wins, otherwise Fred wins.

If Alexis held up a fingers and Fred held up b fingers, determine who won the game?

Constraints:

$1 \leq T \leq 1000$

$1 \leq a, b \leq 5$

Input Format:

- First line will contain T, number of testcases. Then the T testcases follow.
- Each testcase contains of a single line of input, two integers a,b which denote the number played by Alexis and Fred respectively.

Output Format:

For each testcase, output which player won the game.

Code:

```
import java.util.*;
```

```
public class Class815427160771 {
```

```
    public static void main (String[] args){
```

```
        Scanner sc= new Scanner(System.in);
```

```
        int a,b,d;
```

```
        int t= sc.nextInt();
```

```
        while(t-->0)
```

```
        {
```

```
            a= sc.nextInt();
```

```
            b= sc.nextInt();
```

```
            d=(a+b)%2;
```

```
            if(d==0)
```

```
            {
```

```
                System.out.println("Fred");
```

```
            }
```

```
        else
```

```

        {
            System.out.println("Alexis");
        }
    }
}

```

✓ Logical Test Cases

Test Case 1

INPUT (STDIN)

```

2
1 1
1 2
            
```

EXPECTED OUTPUT

```

Fred
Alexis
            
```

Test Case 2

INPUT (STDIN)

```

4
1 2
2 2
2 4
4 8
            
```

EXPECTED OUTPUT

```

Alexis
Fred
Fred
Fred
            
```

100. Question description

Finally, University has started calling students back to campus. There are so many students and thus due to some safety measures the University can't call back all the students on the same day. It currently has the capacity of screening K students on a single day. There is a total of N students. What's the minimum number of days required for all the students to be back on the campus?

Constraints:

$1 \leq N, K \leq 100$

Input Format:

The first and only line of each test case contains two space-separated integers N,K.

Output Format:

Print a single line containing one integer - the minimum number of days required for all the students to be back on the campus.

Code:

```
import java.util.Scanner;

public class Class815427160771 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int N = input.nextInt();

        int K = input.nextInt();

        if(N%K==0)

        {

            System.out.println("1");

        }

        else if(N%K!=0){

            System.out.println(Math.round(N/K)+1); } } }
```

✓ Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
3 3	3 2
EXPECTED OUTPUT	EXPECTED OUTPUT
1	2