Tagline:

Description:

Scoring:

Max points per question will vary on the difficulty of the question.

You can provide your submission in Python, Java, C++, JavaScript.

Time Duration: 3 hours

Rules:

- This is an IOI style contest. This means that the problems will be partially graded. You will get score for passing certain test data.
- The details of the failed test cases will also be visible on your solution page.
- You can submit solutions as many times as you'd like, there are no penalties for incorrect submissions. Only your best correct submission will be considered.
- Those who achieve the score first will be placed higher in the ranklist in case of a tie.
- You can also send in your queries in an email to divyansh@codeyoung.com, during the contest.
- Please do not discuss strategy, suggestions or tips in the comments during a live contest. Posting questions clarifying the problem statement is ok. If you are unsure, email us at divyansh@codeyoung.com .

What's in it for you?

The idea behind these programming contests is that we want you to learn while competing. Also we believe that it is alright to refer to tutorials, books and other materials, learn a concept and then apply the same to solve a problem during a contest. But it is not alright to copy other people's solutions or seek other people's help to solve a problem without understanding it. The dividing line may seem to be thin but it can be captured by the spirit of learning. If whatever you are doing is making you learn while you do so, we tend to believe that it is alright. However, all the participants are expected to abide to Code Of Conduct.

Not just that, as a token of motivation we also give away cash prizes and goodies to the winners of the contest. However, our sole intention lies in making our users learn new concepts while competing. The prizes will be given according the ranklist:

1st Rank: INR

2nd Rank: INR

3rd Rank: INR

Problems:

01. You are given the height H (in metres) and mass M (in kilograms) of youngCoder. The Body Mass Index (BMI) of a person is computed as M / H^2 (M divided by square of H).

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

- Category 1: Underweight if BMI ≤18
- Category 2: Normal weight if BMI ∈{19, 2020,……, 24}
- Category 3: Overweight if BMI ∈{25, 2626,……, 29}
- Category 4: Obesity if BMI ≥30

Input:

- A single line of input, with two space separated integers, M and H, which denote the mass and height of youngCoder respectively.

Output:

Output in a single line, 1,2, 3 or 4, based on the category in which youngCoder falls.

Constraints

- $1 \le T \le 2 * 10^4$
- $1 \le M \le 10^4$
- $1 \le H \le 10^2$
- Its guaranteed that H^2 divides M.

Sample Input:

72 2

80 2

120 2

Sample Output:

1

2

4

Explanation:

Case 1: Since $M / H^2 = 18$, therefore person falls in category 1.

Case 2: Since $M / H^2 = 20$, therefore person falls in category 2.

Case 3: Since $M / H^2 = 30$, therefore person falls in category 4.

---

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

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

Input

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

Output

Print a single line containing the string "YES" if it is possible for YoungCoder to wear two socks with the same colour or "NO" if it is impossible (without quotes).

Constraints

- $1 \leq A,B,C \leq 10$

Example Input 1

5 4 3

Example Output 1

NO

Explanation

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

Example Input 2

5 5 5

Example Output 2

YES

Explanation

Since all three socks have the same colour, YoungCoder can wear a pair of socks with the same colour.

03. Suppose YoungCoder is stuck on an island and currently he has xx units of food supply and y units of water supply in total that he could collect from the island. He needs x_r units of food supply and y_r units of water supply per day at the minimal to have sufficient energy to build a boat from the woods and also to live for another day. Assuming it takes exactly DD days to build the boat and reach the shore, tell whether YoungCoder has the sufficient amount of supplies to be able to reach the shore by building the boat?

Input:

- Line of input, five integers x ,y , x_r ,y_r ,D

Output:

For each testcase, output in a single line answer "YES" if YoungCoder can reach the shore by building the boat and "NO" if not (without quotes).

Constraints

- 1≤T≤300
- 1≤x,y≤100
- 1≤xr,yr,D≤10

Sample Input

4 2 1 1 1

4 2 1 3 1

4 2 4 2 2

Sample Output:

YES

NO

NO

Explanation:

TestCase 1: YoungCoder's food supply will last for 4/1=4 days and water supply will last for 2/1=2 days, so in total he will be able to survive for min(4,2)=2 days and since required time to reach the shore is 1 day, he will be able to reach there.

TestCase 2: YoungCoder's food supply will last for 4/1=4 days and water supply will last for 2/3=0.67 days, so in total he will be able to survive for min(4,0.67)=0.67 days and since required time to reach the shore is 1 day, he won't be able to reach there.

TestCase 3: YoungCoder's food supply will last for 4/4=1 day and water supply will also last for 2/2=1 day, so in total he will be able to survive for min(1,1)=1 day and since required time to reach the shore is 2 days, he won't be able to reach there.

---

04. YoungCoder is a big fan of Coldplay. Every Sunday, he will drive to a park taking M minutes to reach there, and during the ride he will play a single song on a loop. Today, he has got the latest song which is in total S minutes long. He is interested to know how many times will he be able to play the song completely.

Input

- Two space-separated integers M, S- the duration of the trip and the duration of the song, both in minutes.

Output

For each test case, output in a single line the answer to the problem.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq M \leq 100$
- $1 \leq S \leq 10$

Sample Input

10 5

10 6

9 10

Sample Output

2

1

0

Explanation

- **Test case 1:** YoungCoder listens to the song once from 0−5 minutes and next from 5−10 minutes.
- **Test case 2:** YoungCoder listens to the song from 0−6 minutes but now he has only 4 minutes left so he can't complete the song again.
- **Test case 3:** Since the song lasts longer than the journey, YoungCoder won't be able to complete it even once.

05.CodeYoung is holding a virtual marathon for the categories 10 km, 21km and 42 km having prizes A,B,C (A<B<C) respectively to promote physical fitness. In order to claim the prize in a particular category the person must cover the total distance for that category within D days. Also a single person **cannot** apply in multiple categories.

Given the maximum distance d km that YoungCoder can cover in a single day, find the maximum prize that YoungCoder can win at the end of D days. If YoungCoder can't win any prize, print 0.

Input

- Each test case contains a single line of input, five integers D,d,A,B,C.

Output

For each test case, output in a single line the answer to the problem.

Constraints

- $1 \le T \le 50$
- $1 \le D \le 10$
- $1 \le d \le 5$
- $1 \le A < B < C \le 10^5$

Sample Input

1 1 1 2 3

10 1 1 2 3

10 3 1 2 3

Sample Output

0

1

2

Explanation

Test Case 1**:** The maximum distance covered by YoungCoder is 1·1=1km which is less than any of the available distance categories. Hence YoungCoder won't be able to claim a prize in any of the categories.

Test Case 2**:** The maximum distance covered by YoungCoder is 10·1=10 km which is equal to distance of the first category. Hence YoungCoder can claim the maximum prize of 1 units.

Test Case 3**:** The maximum distance covered by YoungCoder is 10·3=30 km which is more than the distance of the second category but less than that of the third category. Hence YoungCoder can claim the maximum prize of 2 units.

---

06. Suppose the population is divided into the age groups [1,10],[11,20],[21,30],…,[91,∞). The age groups are numbered from 1 to 10 and there are Xi people in age group 'I' .The COVID vaccine drive has started and people will be vaccinated in the decreasing order of their age groups. Suppose P people are vaccinated per day and if less than P people are left in an age group, then the remaining doses for that day are given to the people of immediate lower age group and the process for that day continues until the vaccines for that day are finished or the entire population has been vaccinated. The selection of a person from a given age group is done randomly.

Given X, P, and YoungCoder's age group G, tell the minimum and the maximum number of days will it take for YoungCoder to get vaccinated.

Input

- Each test case contains a single line of input, twelve integers G, P, X1,X2,…, X10.

Output

For each test case, output in a single line two space-separated integers, the minimum and maximum time required for YoungCoder to get vaccinated.

Constraints

- $1 \le T \le 10^4$
- $1 \le G \le 10$
- $1 \le P, Xi \le 10^5$

Sample Input

5 2 2 2 2 2 2 2 2 2 2

5 2 2 2 2 2 2 3 2 2 2 2

9 4 2 2 2 2 3 2 2 2 2

Sample Output

6 6

6 7

1 1

Explanation

Test Case 1**:** It takes one day for every age group to get vaccinated.

Test Case 2**:** Age groups 10−7 get vaccinated in a total of 4 days. On the 5-th day, 2 out of 3 people of age group 6 get vaccinated. On the 6-th day, the last person of the 6-th age group and one random person of the 5-th age group gets vaccinated. So if the person is YoungCoder, the minimum time required for him to get vaccinated will be 6 days, and if not, YoungCoder will get vaccinated on the 7-th day.

Test Case 3**:** The last 2 age groups get vaccinated on the same day.

07

Given a string s, return *the string after replacing every uppercase letter with the same lowercase letter*.

Input

- Each test case contains a single line of input, a string with mixed case characters.

Output

For each test case, output a string in lowercase.

Example 1:

Input: s = "Hello"

Output: "hello"

Example 2:

Input: s = "here"

Output: "here"

Example 3:

Input: s = "LOVELY"

Output: "lovely"

Constraints:

- 1 <= s.length <= 100
- s consists of printable ASCII characters.

Think ASCII!

08

Given an integer **N**, the task is to check if the given number **N** is even or odd. If it is found to be even, then print **"Even"**. Otherwise, print **"Odd"**.

Input

- Each test case contains a single input of integer n.

Output

For each test case, output Even or Odd.

You can consider 0 as even.

Examples:

Input: *N = 2*
Output: *Even*

Input: *N = 5*
Output: *Odd*

Constraints: 1<n<1000

09

youngCoder likes the number **4** very much.

Impressed by the power of this number, youngCoder has begun to look for occurrences of four anywhere. He has a list of **T** integers, for each of them he wants to calculate the number of occurrences of the digit **4** in the decimal representation. He is too busy now, so please help him.

Input

The first line of input consists of a single integer **T**, denoting the number of integers in youngCoder's list.

Then, there are **T** lines, each of them contain a single integer from the list.

Output

Output **T** lines. Each of these lines should contain the number of occurrences of the digit **4** in the respective integer from youngCoder's list.

Constraints

- **1 ≤ T ≤ 10^3**

Example

Input:

5

447474

228

6664

40

81

Output:

4

0

1

1

0

10

Write a program to take two numbers as input and print their difference if the first number is greater than the second number otherwise print their sum.

Input:

- First line will contain the first number (N1)
- Second line will contain the second number (N2)

Output:

Output a single line containing the difference of 2 numbers (N1−N2) if the first number is greater than the second number otherwise output their sum (N1+N2).

Constraints

- −1000≤N1≤1000
- −1000≤N2≤1000

Sample Input:

82

28

Sample Output:

54

Solutions: