

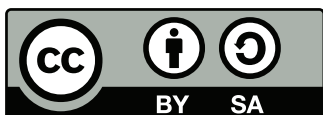
# FRESHMEN PROGRAMMING CONTEST

## CONTEST PROBLEM SET

March 24, 2016



- A A Match of Table Tennis
- B Breaking the Cipher
- C Crawling
- D Debug
- E Expensive Floor
- F Fences
- G Guessing Game
- H Helping Out



Copyright © 2016 by the FPC 2016 Jury.

Licensed under the Creative Commons Attribution-Share Alike license version 3.0:

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

## A - A Match of Table Tennis

The rules of a match of table tennis are quite easy. When a match is played best of five games, it means that the player who wins three games wins. Since there is no point of playing after a winner is decided only the following results are possible: 3-0, 3-1, 3-2, 0-3, 1-3 and 2-3. For each game, the first player to reach 11 points wins that game. However, a game must be won by at least a two point margin. Therefore scores such as 9-12 and 17-14 are not possible, but 9-11, 12-10 and 4-8 are. Each player serves two points in a row and then switch server. However, if a score of 10-10 is reached in any game, then each server serves only one point before the server is switched. In the first game the guest player starts serving, the next game the host starts serving, ect. Both players can score a point, not just the server.

While playing a match of table tennis players are often confused and don't know who the server is. You should help the table tennis players with a program, which can decide who the server is based on the scores. Since people can make mistakes while giving input to the program, you should also check for consistency. Note that you can only start a new game, when the previous game has a winner.

### Input

On the first line a single integer  $n$ ,  $1 \leq n \leq 5$ , the number of games played. On the second line  $n$  scores in the form  $a - b$ , with  $0 \leq a, b \leq 99$ .

### Output

Output "HOST" when the host player is serving, "GUEST" when the guest player is serving or "INCONSISTENT" when the input is not correct.

## Examples

input 1	output 1
1 4-5	GUEST
input 2	output 2
4 1-11 2-11 3-11 4-11	INCONSISTENT
input 3	output 3
2 4-8 3-10	INCONSISTENT
input 4	output 4
4 3-11 4-11 11-9 12-10	HOST

## B - Breaking the Cipher

It has already been years since Alice moved into her new house in the countryside, but only a week ago she suddenly found a mysterious safe, requiring a secret combination to unlock, hidden in her basement. She jumped onto the internet to ask strangers for clues and she received a message from a mysterious man called Bob. He included the combination to Alice's safe, but noted that he had encrypted it with the RSA algorithm using Alice's public key. On the internet you find that the RSA algorithm works as follows:

**Key generation:**

1. Choose two distinct prime numbers  $p$  and  $q$ .
2. Compute  $n = pq$ .
3. Compute  $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1) = n - (p+q-1)$ , where  $\phi$  is Euler's totient function.
4. Choose an integer  $e$  such that  $1 < e < \phi(n)$  and  $\gcd(e, \phi(n)) = 1$ ; i.e.,  $e$  and  $\phi(n)$  are coprime.
5. Determine  $d$  as  $d \equiv e^{-1} \pmod{\phi(n)}$ ; i.e.,  $d \cdot e \equiv 1 \pmod{\phi(n)}$ .

**Encryption:**

Suppose that Bob would like to send message  $M$  to Alice. He then computes the ciphertext  $C$ , using Alice's public key  $e$ , corresponding to

$$C \equiv M^e \pmod{n}$$

**Decryption:**

Alice can recover  $M$  from  $C$  by using her private key exponent  $d$  by computing

$$M \equiv C^d \pmod{n}$$

Additionally, Alice has learned that the following congruence could prove to be useful for the decryption process:

$$(a \cdot b) \pmod{n} \equiv ((a \pmod{n}) \cdot (b \pmod{n})) \pmod{n}$$

Alice has already chosen the integers  $p$ ,  $q$ , and  $e$  accordingly and needs your help to decrypt the message  $C$  she has received from Bob.

### Input

The input to this problem is structured as follows: The first line contains three integers,  $1 < p, q, e < 1000$ , respectively. The second line contains one integer  $C$ , the encrypted secret combination to Alice's safe.

## Output

One line with the (decrypted) combination  $M$  to Alice's safe.

## Examples

input 1	output 1
751 337 739 735	15075
input 2	output 2
181 677 739 567	11662
input 3	output 3
467 823 397 802	368133

## C - Crawling

Saint Alex is a swimming coach that trains his students to participate in a swimming tournament held in the FPC (Forbidden Pool for Crawling). In this tournament, participants must use the front crawl to swim to the other side of the FPC and then use the back crawl to return. Since this tournament is the most prestigious one on the planet, Alex considers winning this tournament the crown jewel of his career.

Being desperate to win, he spends a lot of time watching practice sessions held by other coaches and noting down the times of the students. Unfortunately, he forgot to note down the time of his own students, a crucial piece of information if he wants to know if any of his students makes a chance at all at winning the competition! Luckily, he found the results of a bachelor thesis group that measured the speeds of his students when using the front crawl and the back crawl during the final practice session. Since his number of students can be rather large (up to 10.000 students), it would take too long to check by hand if there is a student that can beat the fastest student of the rival coaches. To make matters even worse, the size of the FPC also changes every year! While it is quite the mystery how he fits and manages 10.000 students in a swimming pool during the practice session, not to mention how that bachelor thesis group managed to register them all, he asks you to help him out.

You are given the task to check – given the fastest time Alex noted down during his observations – if there is a student that can beat the fastest student of the rival coaches.

For the sake of simplicity, we assume that every student swims in a perfectly straight line (thus the width of the FPC is irrelevant, don't worry it's still wide enough for everyone) and there is no delay when switching between the front and back crawl when the student reaches either side of the FPC. Furthermore, every student swims at a constant speed.

### Input

The input is as follows. First, an integer  $l$ ,  $20 \leq l \leq 1000$  the length of the FPC in meters. Next, an integer  $n$ ,  $1 \leq n \leq 10000$  the number of students Alex is coaching. Then, a float  $t$ ,  $2.0 \leq t \leq 3600.0$  the time of the fastest student Alex has observed in seconds. After this,  $n$  lines follow, each line containing two space separated float values  $f$  and  $b$   $0.0 < f, b \leq 1000.0$ , the speed of the student when doing the front crawl and the back crawl in m/s (meter per second), respectively.

### Output

Output "HOPE" if there is at least one student that can beat time  $t$ , else output "DOOMED".

## Examples

input 1	output 1
2 2 4.0 0.42 0.21 0.24 0.332	DOOMED
input 2	output 2
2 2 4.0 1.0 1.0 1.4 0.6	DOOMED
input 3	output 3
2 2 4.0 2.0 2.0 1.498 0.642	HOPE



## D - Debug

You just joined a new team to participate in the FPC, because the previous member left the team frustrated. Frustrated that he could not beat a time limit<sup>1</sup> of a certain problem of an earlier FPC. To show you indeed possess the skills to solve problems during a FPC, you decide to solve the problem the other member could not. Unfortunately, he never told anyone which problem he was actually trying to solve. Luckily, you still got his code!

Your task is to debug the problem his code is solving and speed it up. You may assume that in its current state, it is correct but too slow. The (Java) code is given below.

```
import java.util.Scanner;

class D {
    public static int n(int n) {
        int p = 1;
        if (n == 1 || n == 0) p = 0;
        for (int i = 1; i < n; i++) {
            if (i == 1) continue;
            if (n % i == 0) p = 0;
        }
        return p;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        if (n(sc.nextInt()) == 1) System.out.println("yes");
        else System.out.println("no");
    }
}
```

---

<sup>1</sup>This was based on the text "Correct but TIME LIMIT! AARGH!" being scratched in the wall.

## **Input**

As can be seen in the provided code. The only input is an integer  $0 \leq n \leq 10^9$ .

## **Output**

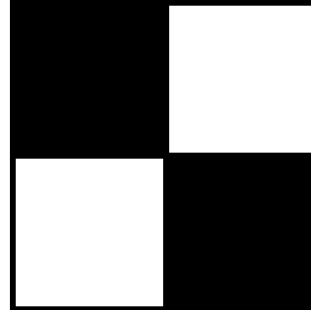
Keep the same output as provided in the code. You only need to speed up things.

## **Examples**

None! Run the code and try to figure out what is implemented.

## E - Expensive Floor

A year ago, your friend John moved to Delft. Recently, he called you to ask your help: he wants to redecorate his house. You spent many hours looking for all kinds of furniture and decorations. Now there's only one thing left: he wants a new floor in his house. You and John decide to visit a store nearby to look for a nice one.



**Figure 1** – A floor with five squares.

After looking a bit around in the shop, John noticed a special floor that has the pattern of a checkerboard. The floor is very special since it is sold in one big piece and not in separate black and white tiles. John is interested in the floor and asks the shop keeper for more information. The size of his floor is  $5 \times 5$  (thus  $25m^2$ ) and a floor in this size will cost him 55 euro. He asks how this price is calculated: it turns out that the floor costs one euro for every square that can be found on the floor. Note that each black or white tile has a size of  $1 \times 1$  meter. So for instance, the figure visible to the right, has a size of  $2 \times 2$  meter. In the figure, we count a total of five squares.

John decides to buy the floor and it looks amazing in his house! You decide to also buy this particular floor but your house is way bigger and you would like to know how much you will have to pay. You decide to write a program to calculate the total price in euro for a given floor of  $n \times n$  meters.

### Input

An integer  $n$  with  $n \leq 10^5$ , representing a  $n \times n$  floor.

### Output

The price in euro you have to pay for the  $n \times n$  floor.

## Examples

input 1	output 1
1	1
input 2	output 2
5	55
input 3	output 3
98583	319368033160804

## F - Fences

Farmer Fred has a farm with a circular-shaped field on which he is keeping flamingos. A few of his flamingos have already been killed by a few foxes, therefore farmer Fred wants to construct fences around the field. Fred knows the area of his field, because his female companion told him. Your job is to create a program that takes as input the area of his circular shaped field in square meters and outputs the total length of the fence needed guard the flamingos from the foxes, in meters. Note that Fred is going to buy his fence from the Fabulous Fences Firm which only sells fences per 10 centimeter.

### Input

A single integer  $f$  ( $1 \leq f \leq 10^{18}$ ), the area of the field in square meters.

### Output

Output the total length of fence needed for the field, in meters.

### Examples

input 1	output 1
12	12.3
input 2	output 2
46	24.1
input 3	output 3
96	34.8

*This page is intentionally left (almost) blank.*

## G - Guessing Game

Some days ago a friend of yours got himself a very powerful computer system. Using chatting software, you ask him what the system costs. He won't tell you the cost, but he will let you guess it once he has assembled his new system.

You have to pick a number, and he will message you back whether your guess is too low, too high or correct. Since you only have a limited amount of time to guess the correct price after he is done assembling, you decide to write a program to read his responses and automatically send your guesses for you.

The input given depends interactively on the output of your program.

### Input

The input consists of a series of words, each line being one of three:

LOWER, your guess was too high.

HIGHER, your guess was too low.

GAME, you guessed correctly

Note: you will have to start with a guess before you will receive any input.

### Output

The output consists of two options:

ASK  $x$ , guess the number  $x$ .

OVER  $y$ , the amount  $y$ ,  $0 \leq y \leq 10^9$ , the system costs. This will end the guessing game.

*This page is intentionally left (almost) blank.*



## H - Helping Out

For the last couple of years gaming competitions have been very popular at the TU Delft. During every competition, any player can obtain up to 100 points. However, the administration of these competitions has been absolutely terrible, because the result of every competition was written down on a piece of paper and all these were scattered across the TU Delft. A group of students has managed to find (almost) all of these lists of scores, but they would like to compute the total score of each participant. Given a list of (possibly duplicate) names and their scores, help the students out by creating a list of the total score of each participant, sorted in **alphabetic order**.

### Input

On the first line an integer  $1 \leq n \leq 10^5$ . On the next  $n$  lines, the name (a string containing no spaces) of person  $M$  followed by an integer  $1 \leq k \leq 100$ , the score of  $M$  in any arbitrary competition.

### Output

The output consists of  $m$  lines, where  $m$  is the number of unique names in the input. Give in alphabetic order the names of all participants and their total score, separated by a space.

### Examples

input 1	output 1
8 Grayson 70 Dante 20 Anton 60 Grayson 73 Grayson 97 Dante 92 Tanisha 56 Grayson 96	Anton 60 Dante 112 Grayson 336 Tanisha 56

input 2	output 2
8 Alexus 36 Jaxon 80 Hali 12 Shayna 31 Xavier 46 Xavier 17 Jaxon 15 Alexus 85	Alexus 121 Hali 12 Jaxon 95 Shayna 31 Xavier 63