Olimpiada Informática Española

Regional de Madrid

Problem set

February 19, 2022

Every great developer you know got there by solving problems they were unqualified to solve until they actually did it
Patrick McKenzie

February 19, 2022

Problem set

A Same Remainder **3**

B Greatest with Two Primes **5**

C Where Do We Build the Airports? **7**

D Amoebas **9**

E Online Games **11**

F 10-Sum **13**

G Worms **15**

H Hyperbinary Representation **17**

Problem setters:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Blanca Huergo Muñoz (Olimpiada Informática Femenina)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Alberto Verdejo (Universidad Complutense de Madrid)

A
Same Remainder

Given natural numbers A, B, D and R, your task is to find how many numbers in the interval $[A..B]$ that when divided by D yield a remainder of R

For example, if $A = 1$, $B = 10$, $D = 5$ and $R = 3$, the answer is 2, as only numbers 3 and 8 fit the description.

Input

Input begins with a line with an integer t, the number of test cases that follow.

There will be one line of input per test case, containing the four numbers A, B, D and R.

Constraints

$0 \leq A \leq B \leq 10^{18}$
$1 \leq D \leq B$
$0 \leq R < D$

Subtasks

1. (5 points) $A = 0$, $B < 10$, $R = 0$
2. (10 points) $A = 0$, $B \leq 10^3$
3. (15 points) $B \leq 10^9$, $B - A \leq 10^3$
4. (50 points) $B \leq 10^9$
5. (20 points) original constraints

Output

For each test case, output one line with one integer: the number of values in the interval $[A..B]$ that, when divided by D, yield a remainder of R.

Sample input

```
4
0 4 2 0
1 10 5 3
0 13 5 3
10 19 2 0
```

Sample output

```
3
2
3
5
```

B
Greatest with Two Primes

All natural numbers have a unique prime factorisation. For example, we have that:

$$1.547 = 7 \times 13 \times 17$$

Some numbers have the unique characteristic of having only two distinct prime factors. For example, 6, 12, 18 and 36 have 2 and 3 as their only prime factors.

Given two prime numbers, p and q, what is the largest number that has only (and exactly) those two prime factors and that is not greater than a third number, n?

Input

The program must process several test cases, each described in one line of input. A test case will consist of three numbers, n, p and q, where p and q are two distinct primes.

The last line of input will consist of a `0` and should not be processed.

Constraints

$1 \leq n,\ p,\ q \leq 1.000.000.000$
$p \neq q$ and both are prime

Subtasks

1. (5 points) $n < p{\cdot}q \leq 1.000.000.000$
2. (35 points) $n < p^2{\cdot}q^2 \leq 1.000.000.00$
3. (60 points) original constraints

Output

For each test case, the program must output the largest number that is not greater than n and which has in its prime factorisation exactly p and q as factors. If there is no such number, output a 0.

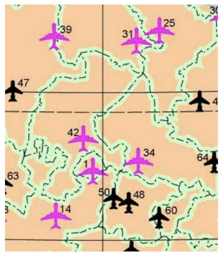
Sample input

```
20 2 3
10 3 2
23 11 7
0
```

Sample output

```
18
6
0
```

C
Where Do We Build the Airports?



In an attempt to attract investors, a nation's Government wants to improve access to one of its more forgotten regions. Several cities in this region must have access to an airport.

An option would be to build an airport in each of these cities, but the Government is looking for options where expenditure is low. In particular, it will consider a city to have access to an airport if they build an airport in it or if it is possible to travel to an airport in another city by car.

The region has received little expenditure in infrastructures in the past and so there are no roads built yet, but if building them reduces the cost of connecting the region, as this will mean less airports required, the Government is open to do so.

Different studies have been carried out regarding the possible roads that could be built, to the point where there are sometimes different roads that could be built between the two same cities, possibly with different costs.

Given the cost of building an airport in each city and a list of possible roads that could be built between pairs of cities, with their respective costs, the Government needs your help now in order to find the cheapest way of guaranteeing that all cities now have access to an airport. The objective is to make this access as easy as possible, so if there are several ways to achieve this minimum cost, pick the one with the most airports.

Input

The input consists of several test cases. Each test case starts with a line with two integers: N, the number of cities, and M, the number of roads that can be built.

Then comes a line with N integers, $A_1 \ldots A_N$, where A_i is the cost of building an airport in the i'th city.

The next M lines describe the possible roads: every line has three integers X, Y and C, where X and Y are two cities and C is the cost of building a road between X and Y.

Input ends with two zeroes, `0 0`.

Constraints

$1 \leq N \leq 200.000$
$0 \leq M \leq 200.000$
$1 \leq A_i \leq 1.000.000, \sum A_i \leq 10^9$
$1 \leq X, Y \leq N$
$1 \leq C \leq 100.000$
If t is the number of test cases, $\sum_{i=1}^{t} N_i + M_i \leq 400.000$

Subtasks

1. (5 points) $N = 2$, $M \leq 1$ and all the A_i are the same
2. (5 points) $N = 3$, $M = 2$ and all the A_i are the same
3. (25 points) $N \leq 100$, $M \leq 1.000$ and all the A_i are the same
4. (40 points) all the A_i are the same
5. (25 points) original constraints

Output

Output one line per test case with two integers: the minimum cost required to build the roads and airports necessary to connect all cities to at least one airport and the number of airports in that plan. Remember that if there are several plans with this minimum cost, the one with the largest number of airports shall be chosen.

Sample input

```
4 4
100 100 100 100
1 2 10
2 3 30
4 1 40
4 3 20
5 3
1000 2000 1000 1000 3000
4 5 10
1 2 20
3 2 30
0 0
```

Sample output

```
160 1
2060 2
```

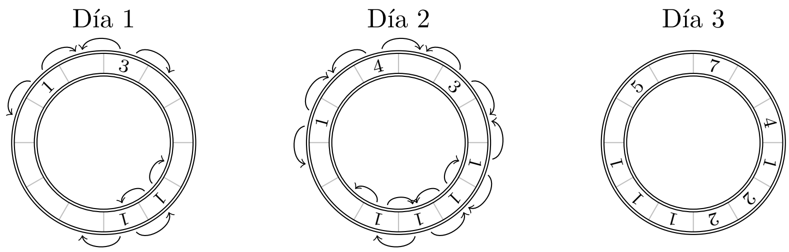
D
Amoebas

Cellular mitosis is the reproductive process where the nucleus of a eukaryotic cell splits in two nuclei, keeping the genetic information the original nucleus stored intact in both the new nuclei. Mitosis is then followed by cellular division, meaning that where there was originally a single cell, there are now two.

Mitosis is essential, for example, for growth or tissue repair and is the basis of the reproduction of simple organisms comprised of one or few cells. This is the case of amoebas, unicellular organisms that reproduce asexually: when the time arrives, the genetic material, the cell elongates, citoplasm splits in two, and instead of one cell we now have two.

In a lab we have a population of amoebas, where several "plots" are distributed around ring. Initially, we have a fixed number of amoebas in each plot, which grow throughout and duplicate once per day. Every time an amoeba splits in two, one of the copies moves to the plot to the right of the current plot, whereas the other one moves to the plot to the left of the current plot.

The figure shows an example of the evolution of a population for 3 days. On the first day, there are six amoebas distributed in four plots (omitting the number in empty plots). These amoebas duplicate and move, as explained above, such that at the start of the second day there are now twelve and at the start of the third day 24.



Given the number of amoebas in each plot of a population, how many amoebas will there be in each plot at the end of a specific given day?

Input

Input will consist of t test cases, each one taking two lines.

The first line of each test case will contain two numbers, n and d indicating, respectively, the number of plots and the day we are asking about. The second line will contain n integers: the number of amoebas in each plot on day 1 (at most 10^9).

Remember that as plots are distributed around a ring, the first plot in the given list has the last plot in the given list as its left neighbour.

The last test case will be followed by a line with two zeroes that should not be processed.

Constraints

$3 \le n \le 1000$, $1 \le d \le 10^9$

Subtasks

1. (3 points) $n \le 5$, $d \le 2$, $t \le 2000$ and plots with up to 1000 initial amoebas
2. (2 points) $n \le 5$, $d \le 2$, $t \le 2000$
3. (20 points) $d \le 100$, $t \le 10$
4. (50 points) $n \le 100$, $t \le 2$
5. (25 points) original restrictions (with $t \le 2$)

Output

For each test case write one line with n integers, indicating the number of amoebas on each plot the last day. As there can be many, instead of the total quantities, output their remainder when divided by 10^9+7.

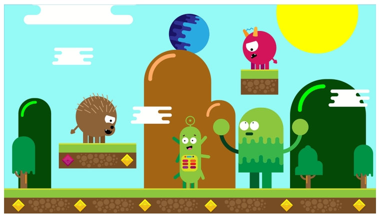
Sample input

```
4 2
0 2 0 0
4 2
2 0 0 0
12 3
1 0 3 0 0 0 1 1 0 0 0 0
0 0
```

Sample output

```
2 0 2 0
0 2 0 2
5 0 7 0 4 1 2 2 1 1 1 0
```

E
Online Games



My grandfather is a big fan of online gaming. At night time, after dinner, he chooses which game he will play for a bit. He then plays one or more games, without changing the game he plays, until he goes to bed. He enjoys this so much that he can remember which game he played each night. I sometimes ask him if he remembers when it was that he chose to play Minecraft for the third time and he soon replies: "I played for the third time on day 15". For my grandfather, day 1 is the day he played an online game for the first time, something he has never spent a night without doing since then. Online games have changed his life!

Could you help the grandfather remember when he has played each game when he no longer has such good memory?

Input

The input of this problem is comprised of several test cases. Each case begins with a line with an integer N: the number of nights the grandfather has played after dinner. In the next line you will see the name of the videogame he played each night. Each name is a chain of no more than 20 characters, with no spaces.

You will then receive an integer M, the number of queries you will have to answer. The next M lines contain these queries, each of them comprised of an integer k and the name of a game `Juego`, which represent the question "When was the k-th day you played the game `Juego`?"

Input ends with a `0`.

Constraints

$1 \leq N \leq 100.000$
$1 \leq M \leq 100.000$
$1 \leq k \leq N$

Subtasks

1. (5 points) the grandfather only plays `LoL`
2. (5 points) $M = 1$, $k = 1$
3. (10 points) $N \leq 100$, $M \leq 100$
4. (15 points) the grandfather only plays `LoL` and `Minecraft`
5. (20 points) the grandfather plays at most 20 different games
6. (45 points) original constraints

Output

For each test case, output a line per question with the day in which the grandfather played the game for the k-th time. Remember that the first day the grandfather played a game is considered day 1. If he has never played that game, or not that many times, output `NO JUEGA`.

After each test case, write three hyphens: `---`.

Sample input

```
7
Pacman Parchis Parchis SuperMario Pacman Parchis Minecraft
5
2 Pacman
1 Minecraft
3 Parchis
1 Oca
2 SuperMario
0
```

Sample output

```
5
7
6
NO JUEGA
NO JUEGA
---
```

F
10-Sum

Izan is tired of getting annoyed whenever he plays *10-Sum*, a very addictive game everyone he knows is playing. The game starts with a list of n integers, all of them from 1 to 10. In each move, you can remove two numbers of the list of they are both equal or if they add up to 10. The objective is to remove all the numbers of the list, obtaining an empty list.

He has also bought the newest version of the game, *S-Sum*, where you have to try to add up to the quantity S, instead of 10. That is, two numbers in the list can be paired up and removed if they are the same or they add up to S.

Given an initial list A, would you be able to tell Izan if there is a solution? That way, he would stop worrying if he already knew that with that initial list no victory is possible.

Input

The input for this problem starts with a line with an integer t: the number of games Izan plays, where each game is described in two lines.

The first line of each game contains two integers: the number S that pairs should add up to and n, the length of A. The second line contains n integers, $A_1 \dots A_n$.

Constraints

$10 \leq S \leq 100$
$1 \leq n \leq 10^6$
$1 \leq A_i \leq S$

Subtasks

1. (3 points) $S = 10$ and n will be an odd number
2. (10 points) $S = 10$ and all positions of A will contain the same number
3. (10 points) $S = 10$ and $A_i = 5$ or $A_i = 10$ for all i
4. (15 points) $S = 10$ and $n \leq 10$
5. (20 points) $S = 10$ and $n \leq 1000$
6. (42 points) original constraints

Output

For each game, output a line with a sentence describing the result: if there exists some way of leaving list A empty, output `SE PUEDE, IZAN!` and otherwise output `NO MERECE LA PENA :(`.

Sample input

```
2
10 6
2 7 4 3 4 8
20 1
7
```

Sample output

```
SE PUEDE, IZAN!
NO MERECE LA PENA :(
```

G
Worms

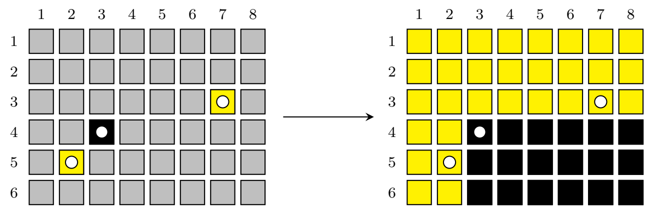
Our peaceful village has been dedicated to agriculture for centuries. More specifically, we plant sunflowers. In fact, legend has it that we were one of the first regions to cultivate them after their arrival to Europe.

However, recently our land has been affected by some small worms which bite the stems of our plants and stop them from turning to face the Sun.

The worms' way of attacking is always the same. Worms get into a crop through the lower right corner (if we look at the crop from above) and extend, forming a rectangle. The result is that when a photo of the land is made, you can observe a rectangular section in the lower right section of plants that do not face the Sun.

In order to evaluate damage, the Town Hall has bought a drone that we can send to a specific coordinate of the crop to see if in that area plants have turned to face the Sun. What we now need to decide is where we are sending the drone to see to what extent an area has been affected.

For example, if we have a crop of dimension 8×6 and we send the drone to coordinates (3,4), (7,3) and (2,5) and it only detects worms in the first of these, then we will certainly know that the affected area runs from position (3, 4) to the lower right corner. The next figure illustrates this example marking healthy areas in yellow and areas with worms in black:



Our job consists of creating an app that helps the Town Hall technician to decide where to send the drone to find out fast the position of the upper left corner of the rectangle affected by worms in several crops. The app will receive the size of the crop and will then ask the technician to send the drone to different coordinates. Depending on what the drone sees in each position, the app will decide where to send it next.

The exact workings of the drone are as follows. When turned on, the app will wait for the technician to introduce the size of the plot that the app is expected to analyse next (two integers, `TamX` and `TamY`). Then, it will ask the technician to send the drone to the coordinates it finds convenient to observe the worm situation in that area. For that, it will write in separate lines commands int the following format: "`? posX posY`". The technician will then send the drone to each of these positions and, every time the drone comes back, will write down a `0` with his keyboard if the area is intact and a `1` if the area has been affected by the worms. When the app is certain about where the upper left corner of the attacked area lies, it will write a line with the result: "`=> posX posY`", the coordinates of the required corner; at that point the technician will type in a new crop's size, starting the cycle again, or an empty size ("`0 0`"), which will make the app quit instantly.

Here you are given an example of the app's working. The data introduced by the technician appears in italics, whereas the data written down by the app appears as bold text.

2 3
? 1 1
1
=> 1 1
8 6
? 3 4
1
? 7 3
0

```
? 2 5
0
=> 3 4
0 0
```

Constraints

$1 \leq$ `TamX`, `TamY` $\leq 10^5$

Subtasks

1. (2 points) `TamY` ≤ 10, `TamX` ≤ 10 and the answer is (`TamX`,`TamY`)
2. (10 points) `TamY` $= 1$, `TamX` ≤ 10
3. (4 points) `TamY` $= 2$, `TamX` ≤ 10
4. (40 points) `TamY` $= 1$, `TamX` $\leq 10^5$
5. (4 points) `TamY` $= 2$, `TamX` $\leq 10^5$
6. (40 points) original constraints

Notes

This is an *interactive* problem. It is therefore important that every time you output something that the grader has to read (that is, every time your app writes something for the technician), you "*flush*" flush the output buffer. This is different for each programming language. In the ones we admit at this competition, it is done as follows:

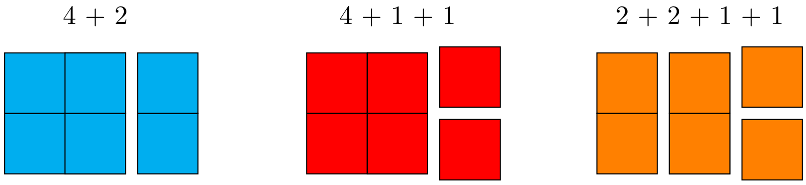
- C++: `cout << endl;` o `cout << flush;`.
- Java: `System.out.flush();`.
- Python: `stdout.flush()`.

H
Hyperbinary Representation

It is a very well-known fact that any integer can be written down uniquely as the sum of powers of two, where each power appears at most once. In fact, the powers used in that sum are the ones that lead to the representation in base 2 (binary) of the number.

For example, $6 = 4 + 2$, which makes the binary representation of 6 equal 110_2.

The *hyperbinary representation* of a number is that in which the number is expressed as the sum of powers of two in which each power, 2^k, appears at most twice. By allowing more than one appearance of the same power, the representation is no longer unique. For example, 6 can be represented in three different ways:



Given a positive number, would you be able to calculate how many hyperbinary representations it has?

Input

The input starts with a line with the number of test cases that follow (at most 2,000). After it come the test cases, where each test case takes up one line and this line contains the number in question: a positive integer n between 1 and 10^{18}.

Constraints

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

Subtasks

1. (2 points) $n \leq 6$
2. (9 points) $n \leq 14$
3. (4 points) $n \leq 1023, n = 2^k - 1$
4. (45 points) $n \leq 1023$
5. (20 points) $n \leq 10^9$
6. (20 points) original constraints

Output

For each test case, write a line with the number of hyperbinary representations the given integer has.

Sample input

```
4
1
2
3
6
```

Sample output

```
1
2
1
3
```