Codeforces Round #643 (Div. 2)

A. Sequence with Digits

1 second, 256 megabytes

Let's define the following recurrence:

$$a_{n+1} = a_n + minDigit(a_n) \cdot maxDigit(a_n).$$

Here $minDigit(x)$ and $maxDigit(x)$ are the minimal and maximal digits in the decimal representation of x without leading zeroes. For examples refer to notes.

Your task is calculate a_K for given a_1 and K.

Input

The first line contains one integer t $(1 \le t \le 1000)$ — the number of independent test cases.

Each test case consists of a single line containing two integers a_1 and K $(1 \le a_1 \le 10^{18}, 1 \le K \le 10^{16})$ separated by a space.

Output

For each test case print one integer a_K on a separate line.

input

```
8
1 4
487 1
487 2
487 3
487 4
487 5
487 6
487 7
```

output

```
42
487
519
528
544
564
588
628
```

$a_1 = 487$

$a_2 = a_1 + minDigit(a_1) \cdot maxDigit(a_1) = 487 + \min(4, 8, 7) \cdot \max(4, 8, 7) = 487 + 4 \cdot 8 = 519$

$a_3 = a_2 + minDigit(a_2) \cdot maxDigit(a_2) = 519 + \min(5, 1, 9) \cdot \max(5, 1, 9) = 519 + 1 \cdot 9 = 528$

$a_4 = a_3 + minDigit(a_3) \cdot maxDigit(a_3) = 528 + \min(5, 2, 8) \cdot \max(5, 2, 8) = 528 + 2 \cdot 8 = 544$

$a_5 = a_4 + minDigit(a_4) \cdot maxDigit(a_4) = 544 + \min(5, 4, 4) \cdot \max(5, 4, 4) = 544 + 4 \cdot 5 = 564$

$a_6 = a_5 + minDigit(a_5) \cdot maxDigit(a_5) = 564 + \min(5, 6, 4) \cdot \max(5, 6, 4) = 564 + 4 \cdot 6 = 588$

$a_7 = a_6 + minDigit(a_6) \cdot maxDigit(a_6) = 588 + \min(5, 8, 8) \cdot \max(5, 8, 8) = 588 + 5 \cdot 8 = 628$

B. Young Explorers

2 seconds, 256 megabytes

Young wilderness explorers set off to their first expedition led by senior explorer Russell. Explorers went into a forest, set up a camp and decided to split into groups to explore as much interesting locations as possible. Russell was trying to form groups, but ran into some difficulties...

Most of the young explorers are inexperienced, and sending them alone would be a mistake. Even Russell himself became senior explorer not long ago. Each of young explorers has a positive integer parameter e_i — his inexperience. Russell decided that an explorer with inexperience e can only join the group of e or more people.

Now Russell needs to figure out how many groups he can organize. It's not necessary to include every explorer in one of the groups: some can stay in the camp. Russell is worried about this expedition, so he asked you to help him.

Input

The first line contains the number of independent test cases T($1 \le T \le 2 \cdot 10^5$). Next $2T$ lines contain description of test cases.

The first line of description of each test case contains the number of young explorers N $(1 \le N \le 2 \cdot 10^5)$.

The second line contains N integers $e_1, e_2, \ldots, e_N$ $(1 \le e_i \le N)$, where e_i is the inexperience of the i-th explorer.

It's guaranteed that sum of all N doesn't exceed $3 \cdot 10^5$.

Output

Print T numbers, each number on a separate line.

In i-th line print the maximum number of groups Russell can form in i-th test case.

input

```
2
3
1 1 1
5
2 3 1 2 2
```

output

```
3
2
```

In the first example we can organize three groups. There will be only one explorer in each group. It's correct because inexperience of each explorer equals to 1, so it's not less than the size of his group.

In the second example we can organize two groups. Explorers with inexperience 1, 2 and 3 will form the first group, and the other two explorers with inexperience equal to 2 will form the second group.

This solution is not unique. For example, we can form the first group using the three explorers with inexperience equal to 2, and the second group using only one explorer with inexperience equal to 1. In this case the young explorer with inexperience equal to 3 will not be included in any group.

C. Count Triangles

1 second, 256 megabytes

Like any unknown mathematician, Yuri has favourite numbers: A, B, C, and D, where $A \le B \le C \le D$. Yuri also likes triangles and once he thought: how many non-degenerate triangles with integer sides x, y, and z exist, such that $A \le x \le B \le y \le C \le z \le D$ holds?

Yuri is preparing problems for a new contest now, so he is very busy. That's why he asked you to calculate the number of triangles with described property.

The triangle is called non-degenerate if and only if its vertices are not collinear.

Input

The first line contains four integers: A, B, C and D ($1 \le A \le B \le C \le D \le 5 \cdot 10^5$) — Yuri's favourite numbers.

Output

Print the number of non-degenerate triangles with integer sides x, y, and z such that the inequality $A \le x \le B \le y \le C \le z \le D$ holds.

input
```
1 2 3 4
```
output
```
4
```

input
```
1 2 2 5
```
output
```
3
```

input
```
500000 500000 500000 500000
```
output
```
1
```

In the first example Yuri can make up triangles with sides $(1, 3, 3)$, $(2, 2, 3)$, $(2, 3, 3)$ and $(2, 3, 4)$.

In the second example Yuri can make up triangles with sides $(1, 2, 2)$, $(2, 2, 2)$ and $(2, 2, 3)$.

In the third example Yuri can make up only one equilateral triangle with sides equal to $5 \cdot 10^5$.

D. Game With Array

1 second, 256 megabytes

Petya and Vasya are competing with each other in a new interesting game as they always do.

At the beginning of the game Petya has to come up with an array of N positive integers. Sum of all elements in his array should be equal to S. Then Petya has to select an integer K such that $0 \le K \le S$.

In order to win, Vasya has to find a non-empty subarray in Petya's array such that the sum of all selected elements equals to either K or $S - K$. Otherwise Vasya loses.

You are given integers N and S. You should determine if Petya can win, considering Vasya plays optimally. If Petya can win, help him to do that.

Input

The first line contains two integers N and S ($1 \le N \le S \le 10^6$) — the required length of the array and the required sum of its elements.

Output

If Petya can win, print "YES" (without quotes) in the first line. Then print Petya's array in the second line. The array should contain N positive integers with sum equal to S. In the third line print K. If there are many correct answers, you can print any of them.

If Petya can't win, print "NO" (without quotes).

You can print each letter in any register (lowercase or uppercase).

input
```
1 4
```
output
```
YES
4
2
```

input
```
3 4
```
output
```
NO
```

input
```
3 8
```
output
```
YES
2 1 5
4
```

E. Restorer Distance

1 second, 256 megabytes

You have to restore the wall. The wall consists of N pillars of bricks, the height of the i-th pillar is initially equal to h_i, the height is measured in number of bricks. After the restoration all the N pillars should have equal heights.

You are allowed the following operations:

- put a brick on top of one pillar, the cost of this operation is A;
- remove a brick from the top of one non-empty pillar, the cost of this operation is R;
- move a brick from the top of one non-empty pillar to the top of another pillar, the cost of this operation is M.

You cannot create additional pillars or ignore some of pre-existing pillars even if their height becomes 0.

What is the minimal total cost of restoration, in other words, what is the minimal total cost to make all the pillars of equal height?

Input

The first line of input contains four integers N, A, R, M ($1 \le N \le 10^5$, $0 \le A, R, M \le 10^4$) — the number of pillars and the costs of operations.

The second line contains N integers h_i ($0 \le h_i \le 10^9$) — initial heights of pillars.

Output

Print one integer — the minimal cost of restoration.

input
```
3 1 100 100
1 3 8
```
output
```
12
```

input
```
3 100 1 100
1 3 8
```
output
```
9
```

input
```
3 100 100 1
1 3 8
```
output
```
4
```

input
```
5 1 2 4
5 5 3 6 5
```
output
```
4
```

input
```
5 1 2 2
5 5 3 6 5
```
output
```
3
```

F. Guess Divisors Count

2 seconds, 256 megabytes

This is an interactive problem.

We have hidden an integer $1 \le X \le 10^9$. You **don't have to** guess this number. You have to **find the number of divisors** of this number, and you **don't even have to find the exact number**: your answer will be considered correct if its absolute error is not greater than 7 **or** its relative error is not greater than 0.5. More formally, let your answer be ans and the number of divisors of X be d, then your answer will be considered correct if **at least one** of the two following conditions is true:

- $|ans - d| \le 7$;
- $\frac{1}{2} \le \frac{ans}{d} \le 2$.

You can make at most 22 queries. One query consists of one integer $1 \le Q \le 10^{18}$. In response, you will get $gcd(X, Q)$ — the greatest common divisor of X and Q.

The number X is fixed before all queries. In other words, **interactor is not adaptive**.

Let's call the process of guessing the number of divisors of number X a *game*. In one test you will have to play T independent games, that is, guess the number of divisors T times for T independent values of X.

Input

The first line of input contains one integer T $(1 \le T \le 100)$ — the number of games.

Interaction

To make a query print a line "`? Q`" $(1 \le Q \le 10^{18})$. After that read one integer $gcd(X, Q)$. You can make no more than 22 such queries during one game.

If you are confident that you have figured out the number of divisors of X with enough precision, you can print your answer in "`! ans`" format. ans have to be an integer. If this was the last game, you have to terminate the program, otherwise you have to start the next game immediately. Note that the interactor doesn't print anything in response to you printing answer.

After printing a query do not forget to output end of line and flush the output. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

Hacks

To hack, use the following format:

The first line contains one integer T $(1 \le T \le 100)$ — the number of games.

Each of the next T lines contains one integer X $(1 \le X \le 10^9)$ — the hidden number.

So the example has the form

```
2
998244353
4194304
```

input
```
2

1

1

1


1024

1048576

4194304
```
output
```

? 982306799268821872

? 230856864650023977

? 134690134760714371

! 5
? 1024

? 1048576

? 1073741824

! 42
```

Why the limitation for number of queries is 22 exactly? Maybe the problem author is a Taylor Swift fan.

Let's look at the example.

In the first game $X = 998\,244\,353$ is hidden. Would be hard to guess this, right? This number is prime, so the number of its divisors is 2. The solution has made several random queries, and all the responses turned out to be 1 (strange things, not even one of three random numbers is divisible by $998\,244\,353$). It's fare to assume that the hidden number doesn't have many divisors, so the solution has answered 5. Why not. This answer will be considered correct since $|5 - 2| = 3 \le 7$.

In the second game $X = 4\,194\,304 = 2^{22}$ is hidden, it has 23 divisors. The solution has made queries $1024 = 2^{10}$, $1\,048\,576 = 2^{20}$, $1\,073\,741\,824 = 2^{30}$ and got responses $1024 = 2^{10}$, $1\,048\,576 = 2^{20}$, $4\,194\,304 = 2^{22}$, respectively. Then the solution got completely confused and answered the answer to The Ultimate Question of Life, the Universe, and Everything. This answer will be considered correct since $\frac{1}{2} \le \frac{42}{23} \le 2$.