Codeforces Beta Round #80 (Div. 2 Only)

A. Blackjack

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One rainy gloomy evening when all modules hid in the nearby cafes to drink hot energetic cocktails, the Hexadecimal virus decided to fly over the Mainframe to look for a Great Idea. And she has found one!

Why not make her own Codeforces, with blackjack and other really cool stuff? Many people will surely be willing to visit this splendid shrine of high culture.

In Mainframe a standard pack of 52 cards is used to play blackjack. The pack contains cards of 13 values: $2, 3, 4, 5, 6, 7, 8, 9, 10$, jacks, queens, kings and aces. Each value also exists in one of four suits: hearts, diamonds, clubs and spades. Also, each card earns some value in points assigned to it: cards with value from two to ten earn from 2 to 10 points, correspondingly. An ace can either earn 1 or 11, whatever the player wishes. The picture cards (king, queen and jack) earn 10 points. The number of points a card earns does not depend on the suit. The rules of the game are very simple. The player gets two cards, if the sum of points of those cards equals n, then the player wins, otherwise the player loses.

The player has already got the first card, it's the queen of spades. To evaluate chances for victory, you should determine how many ways there are to get the second card so that the sum of points exactly equals n.

Input
The only line contains n ($1 \le n \le 25$) — the required sum of points.

Output
Print the numbers of ways to get the second card in the required way if the first card is the queen of spades.

Sample test(s)

input
12
output
4

input
20
output
15

input
10
output
0

Note
In the first sample only four two's of different suits can earn the required sum of points.

In the second sample we can use all tens, jacks, queens and kings; overall it's 15 cards, as the queen of spades (as any other card) is only present once in the pack of cards and it's already in use.

In the third sample there is no card, that would add a zero to the current ten points.

B. Testing Pants for Sadness

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The average miner Vaganych took refresher courses. As soon as a miner completes the courses, he should take exams. The hardest one is a computer test called "Testing Pants for Sadness".

The test consists of n questions; the questions are to be answered strictly in the order in which they are given, from question 1 to question n. Question i contains a_i answer variants, exactly one of them is correct.

A click is regarded as selecting any answer in any question. The goal is to select the correct answer for each of the n questions. If Vaganych selects a wrong answer for some question, then all selected answers become unselected and the test starts from the very beginning, from question 1 again. But Vaganych remembers everything. The order of answers for each question and the order of questions remain unchanged, as well as the question and answers themselves.

Vaganych is very smart and his memory is superb, yet he is unbelievably unlucky and knows nothing whatsoever about the test's theme. How many clicks will he have to perform in the worst case?

Input

The first line contains a positive integer n ($1 \le n \le 100$). It is the number of questions in the test. The second line contains space-separated n positive integers a_i ($1 \le a_i \le 10^9$), the number of answer variants to question i.

Output

Print a single number — the minimal number of clicks needed to pass the test it the worst-case scenario.

Please do not use the %lld specificator to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the %I64d specificator.

Sample test(s)

input
2 1 1
output
2

input
2 2 2
output
5

input
1 10
output
10

Note

Note to the second sample. In the worst-case scenario you will need five clicks:

- the first click selects the first variant to the first question, this answer turns out to be wrong.
- the second click selects the second variant to the first question, it proves correct and we move on to the second question;
- the third click selects the first variant to the second question, it is wrong and we go back to question 1;
- the fourth click selects the second variant to the first question, it proves as correct as it was and we move on to the second question;
- the fifth click selects the second variant to the second question, it proves correct, the test is finished.

C. Cthulhu

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

...Once upon a time a man came to the sea. The sea was stormy and dark. The man started to call for the little mermaid to appear but alas, he only woke up Cthulhu...

Whereas on the other end of the world Pentagon is actively collecting information trying to predict the monster's behavior and preparing the secret super weapon. Due to high seismic activity and poor weather conditions the satellites haven't yet been able to make clear shots of the monster. The analysis of the first shot resulted in an undirected graph with n vertices and m edges. Now the world's best minds are about to determine whether this graph can be regarded as Cthulhu or not.

To add simplicity, let's suppose that Cthulhu looks from the space like some spherical body with tentacles attached to it. Formally, we shall regard as Cthulhu such an undirected graph that can be represented as a set of three or more rooted trees, whose roots are connected by a simple cycle.

It is guaranteed that the graph contains no multiple edges and self-loops.

Input

The first line contains two integers — the number of vertices n and the number of edges m of the graph ($1 \le n \le 100, 0 \le m \le \frac{n \cdot (n-1)}{2}$).

Each of the following m lines contains a pair of integers x and y, that show that an edge exists between vertices x and y ($1 \le x, y \le n, x \ne y$). For each pair of vertices there will be at most one edge between them, no edge connects a vertex to itself.

Output

Print "NO", if the graph is not Cthulhu and "FHTAGN!" if it is.

Sample test(s)

input
6 6 6 3 6 4 5 1 2 5 1 4 5 4
output
FHTAGN!

input
6 5 5 6 4 6 3 1 5 1 1 2
output
NO

Note

Let us denote as a simple cycle a set of v vertices that can be numbered so that the edges will only exist between vertices number 1 and 2, 2 and 3, ..., v - 1 and v, v and 1.

A tree is a connected undirected graph consisting of n vertices and n - 1 edges ($n > 0$).

A rooted tree is a tree where one vertex is selected to be the root.

D. Russian Roulette

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After all the events in Orlando we all know, Sasha and Roma decided to find out who is still the team's biggest loser. Thankfully, Masha found somewhere a revolver with a rotating cylinder of n bullet slots able to contain exactly k bullets, now the boys have a chance to resolve the problem once and for all.

Sasha selects any k out of n slots he wishes and puts bullets there. Roma spins the cylinder so that every of n possible cylinder's shifts is equiprobable. Then the game starts, the players take turns, Sasha starts: he puts the gun to his head and shoots. If there was no bullet in front of the trigger, the cylinder shifts by one position and the weapon is given to Roma for make the same move. The game continues until someone is shot, the survivor is the winner.

Sasha does not want to lose, so he must choose slots for bullets in such a way as to minimize the probability of its own loss. Of all the possible variant he wants to select the lexicographically minimal one, where an empty slot is lexicographically less than a charged one.

More formally, the cylinder of n bullet slots able to contain k bullets can be represented as a string of n characters. Exactly k of them are "X" (charged slots) and the others are "." (uncharged slots).

Let us describe the process of a shot. Suppose that the trigger is in front of the first character of the string (the first slot). If a shot doesn't kill anyone and the cylinder shifts, then the string shifts left. So the first character becomes the last one, the second character becomes the first one, and so on. But the trigger doesn't move. It will be in front of the first character of the resulting string.

Among all the strings that give the minimal probability of loss, Sasha choose the lexicographically minimal one. According to this very string, he charges the gun. You have to help Sasha to charge the gun. For that, each x_i query must be answered: is there a bullet in the positions x_i?

Input

The first line contains three integers n, k and p ($1 \le n \le 10^{18}$, $0 \le k \le n$, $1 \le p \le 1000$) — the number of slots in the cylinder, the number of bullets and the number of queries. Then follow p lines; they are the queries. Each line contains one integer x_i ($1 \le x_i \le n$) the number of slot to describe.

Please do not use the %lld specificator to read or write 64-bit numbers in C++. It is preferred to use cin, cout streams or the %I64d specificator.

Output

For each query print "." if the slot should be empty and "X" if the slot should be charged.

Sample test(s)

input
3 1 3 1 2 3
output
..X

input
6 3 6 1 2 3 4 5 6
output
.X.X.X

input
5 2 5 1 2 3 4 5
output
...XX

Note

The lexicographical comparison of is performed by the < operator in modern programming languages. The a string is lexicographically less that the b string, if there exists such i ($1 \le i \le n$), that $a_i < b_i$, and for any j ($1 \le j < i$) $a_j = b_j$.

E. Time to Raid Cowavans

time limit per test: 4 seconds
memory limit per test: 70 megabytes
input: standard input
output: standard output

As you know, the most intelligent beings on the Earth are, of course, cows. This conclusion was reached long ago by the Martian aliens, as well as a number of other intelligent civilizations from outer space.

Sometimes cows gather into *cowavans*. This seems to be seasonal. But at this time the cows become passive and react poorly to external stimuli. A cowavan is a perfect target for the Martian scientific saucer, it's time for large-scale abductions, or, as the Martians say, raids. Simply put, a cowavan is a set of cows in a row.

If we number all cows in the cowavan with positive integers from 1 to n, then we can formalize the popular model of abduction, known as the (a, b)-*Cowavan Raid*: first they steal a cow number a, then number $a + b$, then — number $a + 2 \cdot b$, and so on, until the number of an abducted cow exceeds n. During one raid the cows are not renumbered.

The aliens would be happy to place all the cows on board of their hospitable ship, but unfortunately, the amount of cargo space is very, very limited. The researchers, knowing the mass of each cow in the cowavan, made p scenarios of the (a, b)-raid. Now they want to identify the following thing for each scenario individually: what total mass of pure beef will get on board of the ship. All the scenarios are independent, in the process of performing the calculations the cows are not being stolen.

Input

The first line contains the only positive integer n ($1 \le n \le 3 \cdot 10^5$) — the number of cows in the cowavan.

The second number contains n positive integer w_i, separated by spaces, where the i-th number describes the mass of the i-th cow in the cowavan ($1 \le w_i \le 10^9$).

The third line contains the only positive integer p — the number of scenarios of (a, b)-raids ($1 \le p \le 3 \cdot 10^5$).

Each following line contains integer parameters a and b of the corresponding scenario ($1 \le a, b \le n$).

Output

Print for each scenario of the (a, b)-raid the total mass of cows, that can be stolen using only this scenario.

Please, do not use the %lld specificator to read or write 64-bit integers in C++. It is recommended to use the cin, cout streams of the %I64d specificator.

Sample test(s)

input
3 1 2 3 2 1 1 1 2
output
6 4

input

```
4
2 3 5 7
3
1 3
2 3
2 2
```

output

```
9
3
10
```