A. Cows and Primitive Roots

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

The cows have just learned what a primitive root is! Given a prime p, a primitive root $\bmod p$ is an integer x $(1 \le x < p)$ such that none of integers $x - 1, x^2 - 1, \ldots, x^{p-2} - 1$ are divisible by p, but $x^{p-1} - 1$ is.

Unfortunately, computing primitive roots can be time consuming, so the cows need your help. Given a prime p, help the cows find the number of primitive roots $\bmod p$.

Input

The input contains a single line containing an integer p $(2 \le p < 2000)$. It is guaranteed that p is a prime.

Output

Output on a single line the number of primitive roots $\bmod p$.

Sample test(s)

input
```
3
```
output
```
1
```

input
```
5
```
output
```
2
```

Note

The only primitive root $\bmod 3$ is 2.

The primitive roots $\bmod 5$ are 2 and 3.

B. Cows and Poker Game

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

There are n cows playing poker at a table. For the current betting phase, each player's status is either "`ALLIN`", "`IN`", or "`FOLDED`", and does not change throughout the phase. To increase the suspense, a player whose current status is not "`FOLDED`" may show his/her hand to the table. However, so as not to affect any betting decisions, he/she may only do so if all other players have a status of either "`ALLIN`" or "`FOLDED`". The player's own status may be either "`ALLIN`" or "`IN`".

Find the number of cows that can currently show their hands without affecting any betting decisions.

Input

The first line contains a single integer, n $(2 \leq n \leq 2 \cdot 10^5)$. The second line contains n characters, each either "`A`", "`I`", or "`F`". The i-th character is "`A`" if the i-th player's status is "`ALLIN`", "`I`" if the i-th player's status is "`IN`", or "`F`" if the i-th player's status is "`FOLDED`".

Output

The first line should contain a single integer denoting the number of players that can currently show their hands.

Sample test(s)

input

```
6
AFFAAA
```

output

```
4
```

input

```
3
AFI
```

output

```
1
```

Note

In the first sample, cows 1, 4, 5, and 6 can show their hands. In the second sample, only cow 3 can show her hand.

C. Cows and Sequence

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bessie and the cows are playing with sequences and need your help. They start with a sequence, initially containing just the number 0, and perform n operations. Each operation is one of the following:

1. Add the integer x_i to the first a_i elements of the sequence.
2. Append an integer k_i to the end of the sequence. (And hence the size of the sequence increases by 1)
3. Remove the last element of the sequence. So, the size of the sequence decreases by one. Note, that this operation can only be done if there are at least two elements in the sequence.

After each operation, the cows would like to know the average of all the numbers in the sequence. Help them!

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of operations. The next n lines describe the operations. Each line will start with an integer t_i $(1 \le t_i \le 3)$, denoting the type of the operation (see above). If $t_i = 1$, it will be followed by two integers a_i, x_i $(|x_i| \le 10^3;\ 1 \le a_i)$. If $t_i = 2$, it will be followed by a single integer k_i $(|k_i| \le 10^3)$. If $t_i = 3$, it will not be followed by anything.

It is guaranteed that all operations are correct (don't touch nonexistent elements) and that there will always be at least one element in the sequence.

Output

Output n lines each containing the average of the numbers in the sequence after the corresponding operation.

The answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}.

Sample test(s)

input

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

output

```
0.500000
0.000000
1.500000
1.333333
1.500000
```

input

```
6
2 1
1 2 20
2 2
1 2 -3
3
3
```

output

```
0.500000
20.500000
14.333333
12.333333
17.500000
17.000000
```

Note

In the second sample, the sequence becomes $\{0\} \to \{0,1\} \to \{20,21\} \to \{20,21,2\} \to \{17,18,2\} \to \{17,18\} \to \{17\}$.

D. Cow Program

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

Farmer John has just given the cows a program to play with! The program contains two integer variables, x and y, and performs the following operations on a sequence $a_1, a_2, ..., a_n$ of positive integers:

1. Initially, $x = 1$ and $y = 0$. If, after any step, $x \le 0$ or $x > n$, the program immediately terminates.
2. The program increases both x and y by a value equal to a_x simultaneously.
3. The program now increases y by a_x while decreasing x by a_x.
4. The program executes steps 2 and 3 (first step 2, then step 3) repeatedly until it terminates (it may never terminate). So, the sequence of executed steps may start with: step 2, step 3, step 2, step 3, step 2 and so on.

The cows are not very good at arithmetic though, and they want to see how the program works. Please help them!

You are given the sequence $a_2, a_3, ..., a_n$. Suppose for each i $(1 \le i \le n - 1)$ we run the program on the sequence $i, a_2, a_3, ..., a_n$. For each such run output the final value of y if the program terminates or `-1` if it does not terminate.

Input

The first line contains a single integer, n $(2 \le n \le 2 \cdot 10^5)$. The next line contains $n - 1$ space separated integers, $a_2, a_3, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output

Output $n - 1$ lines. On the i-th line, print the requested value when the program is run on the sequence $i, a_2, a_3, ...a_n$.

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

Sample test(s)

input
```
4
2 4 1
```
output
```
3
6
8
```

input
```
3
1 2
```
output
```
-1
-1
```

Note

In the first sample

1. For $i = 1,$ x becomes $1 \rightarrow 2 \rightarrow 0$ and y becomes $1 + 2 = 3$.
2. For $i = 2,$ x becomes $1 \rightarrow 3 \rightarrow -1$ and y becomes $2 + 4 = 6$.
3. For $i = 3,$ x becomes $1 \rightarrow 4 \rightarrow 3 \rightarrow 7$ and y becomes $3 + 1 + 4 = 8$.

E. Coin Troubles

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

In the Isle of Guernsey there are n different types of coins. For each i $(1 \le i \le n)$, coin of type i is worth a_i cents. It is possible that $a_i = a_j$ for some i and j $(i \ne j)$.

Bessie has some set of these coins totaling t cents. She tells Jessie q pairs of integers. For each i $(1 \le i \le q)$, the pair b_i, c_i tells Jessie that Bessie has a strictly greater number of coins of type b_i than coins of type c_i. It is known that all b_i are distinct and all c_i are distinct.

Help Jessie find the number of possible combinations of coins Bessie could have. Two combinations are considered different if there is some i $(1 \le i \le n)$, such that the number of coins Bessie has of type i is different in the two combinations. Since the answer can be very large, output it modulo 1000000007 $(10^9 + 7)$.

If there are no possible combinations of coins totaling t cents that satisfy Bessie's conditions, output 0.

Input

The first line contains three space-separated integers, n, q and t $(1 \le n \le 300;\ 0 \le q \le n;\ 1 \le t \le 10^5)$. The second line contains n space separated integers, $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^5)$. The next q lines each contain two distinct space-separated integers, b_i and c_i $(1 \le b_i, c_i \le n;\ b_i \ne c_i)$.

It's guaranteed that all b_i are distinct and all c_i are distinct.

Output

A single integer, the number of valid coin combinations that Bessie could have, modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input
```
4 2 17
3 1 2 5
4 2
3 4
```
output
```
3
```

input
```
3 2 6
3 1 1
1 2
2 3
```
output
```
0
```

input
```
3 2 10
1 2 3
1 2
2 1
```
output
```
0
```

Note

For the first sample, the following 3 combinations give a total of 17 cents and satisfy the given conditions: $\{0\ of\ type\ 1, 1\ of\ type\ 2, 3\ of\ type\ 3, 2\ of\ type\ 4\}, \{0, 0, 6, 1\}, \{2, 0, 3, 1\}$.

No other combinations exist. Note that even though 4 occurs in both b_i and c_i, the problem conditions are still satisfied because all b_i are distinct and all c_i are distinct.