A. DZY Loves Sequences

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

DZY has a sequence a, consisting of n integers.

We'll call a sequence $a_i, a_{i+1}, ..., a_j$ $(1 \le i \le j \le n)$ a subsegment of the sequence a. The value $(j - i + 1)$ denotes the length of the subsegment.

Your task is to find the longest subsegment of a, such that it is possible to change at most one number (change one number to any integer you want) from the subsegment to make the subsegment strictly increasing.

You only need to output the length of the subsegment you find.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The next line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output

In a single line print the answer to the problem — the maximum length of the required subsegment.

Examples

input

```
6
7 2 3 1 5 6
```

output

```
5
```

Note

You can choose subsegment a_2, a_3, a_4, a_5, a_6 and change its 3rd element (that is a_4) to 4.

B. DZY Loves Modification

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

As we know, DZY loves playing games. One day DZY decided to play with a $n \times m$ matrix. To be more precise, he decided to modify the matrix with exactly k operations.

Each modification is one of the following:

1. Pick some row of the matrix and decrease each element of the row by p. This operation brings to DZY the value of pleasure equal to the sum of elements of the row before the decreasing.
2. Pick some column of the matrix and decrease each element of the column by p. This operation brings to DZY the value of pleasure equal to the sum of elements of the column before the decreasing.

DZY wants to know: what is the largest total value of pleasure he could get after performing exactly k modifications? Please, help him to calculate this value.

Input

The first line contains four space-separated integers n, m, k and p ($1 \le n, m \le 10^3$; $1 \le k \le 10^6$; $1 \le p \le 100$).

Then n lines follow. Each of them contains m integers representing a_{ij} ($1 \le a_{ij} \le 10^3$) — the elements of the current row of the matrix.

Output

Output a single integer — the maximum possible total pleasure value DZY could get.

Examples

input

```
2 2 2 2
1 3
2 4
```

output

```
11
```

input

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

output

```
11
```

Note

For the first sample test, we can modify: column 2, row 2. After that the matrix becomes:

```
1 1
0 0
```

For the second sample test, we can modify: column 2, row 2, row 1, column 1, column 2. After that the matrix becomes:

```
-3 -3
-2 -2
```

C. DZY Loves Fibonacci Numbers

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the recurrence relation

$$F_1 = 1;\ F_2 = 1;\ F_n = F_{n-1} + F_{n-2}\ (n > 2).$$

DZY loves Fibonacci numbers very much. Today DZY gives you an array consisting of n integers: $a_1, a_2, ..., a_n$. Moreover, there are m queries, each query has one of the two types:

1. Format of the query "$1\ l\ r$". In reply to the query, you need to add F_{i-l+1} to each element a_i, where $l \le i \le r$.
2. Format of the query "$2\ l\ r$". In reply to the query you should output the value of $\sum_{i=l}^{r} a_i$ modulo $1000000009\ (10^9 + 9)$.

Help DZY reply to all the queries.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 300000$). The second line contains n integers $a_1, a_2, ..., a_n\ (1 \le a_i \le 10^9)$ — initial array a.

Then, m lines follow. A single line describes a single query in the format given in the statement. It is guaranteed that for each query inequality $1 \le l \le r \le n$ holds.

Output

For each query of the second type, print the value of the sum on a single line.

Examples

input

```
4 4
1 2 3 4
1 1 4
2 1 4
1 2 4
2 1 3
```

output

```
17
12
```

Note

After the first query, $a = [2, 3, 5, 7]$.

For the second query, $sum = 2 + 3 + 5 + 7 = 17$.

After the third query, $a = [2, 4, 6, 9]$.

For the fourth query, $sum = 2 + 4 + 6 = 12$.

D. DZY Loves Games

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

Today DZY begins to play an old game. In this game, he is in a big maze with n rooms connected by m corridors (each corridor allows to move in both directions). You can assume that all the rooms are connected with corridors directly or indirectly.

DZY has got lost in the maze. Currently he is in the first room and has k lives. He will act like the follows:

- Firstly he will randomly pick one of the corridors going from his current room. Each outgoing corridor has the same probability to be picked.
- Then he will go through the corridor and then the process repeats.

There are some rooms which have traps in them. The first room definitely has no trap, the n-th room definitely has a trap. Each time DZY enters one of these rooms, he will lost one life. Now, DZY knows that if he enters the n-th room with exactly 2 lives, firstly he will lost one live, but then he will open a bonus round. He wants to know the probability for him to open the bonus round. Please, help him.

Input

The first line contains three integers n, m, k $(2 \le n \le 500;\ 1 \le m \le 10^5;\ 2 \le k \le 10^9)$.

The second line contains n integers, each of them is either 0 or 1. If the i-th number is 1, then the i-th room has a trap, otherwise it has not a trap. **Please note**, that the number of rooms with a trap is no more than 101. It is guaranteed that the first room has no trap, and the n-th room has a trap.

Then m lines follows. Each of them contains two integers u_i, v_i $(1 \le u_i, v_i \le n;\ u_i \ne v_i)$, meaning that current corridor connects two rooms u_i and v_i. It is guaranteed that the corridor system is connected.

Output

Print the only real number — the probability for DZY to open the bonus round. The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-4}.

Examples

input

```
5 5 3
0 0 1 0 1
1 2
2 3
3 4
4 5
1 2
```

output

```
0.25000000
```

input

```
3 2 2
0 1 1
1 2
2 3
```

output

```
-0.00000000
```

input

```
2 1 3
0 1
1 2
```

output

```
1.00000000
```

E. DZY Loves Bridges

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

DZY owns 2^m islands near his home, numbered from 1 to 2^m. He loves building bridges to connect the islands. Every bridge he builds takes one day's time to walk across.

DZY has a strange rule of building the bridges. For every pair of islands u, v $(u \neq v)$, he has built 2^k different bridges connecting them, where $k = \max\{k : 2^k | abs(u - v)\}$ ($a|b$ means b is divisible by a). These bridges are bidirectional.

Also, DZY has built some bridges connecting his home with the islands. Specifically, there are a_i different bridges from his home to the i-th island. These are one-way bridges, so after he leaves his home he will never come back.

DZY decides to go to the islands for sightseeing. At first he is at home. He chooses and walks across one of the bridges connecting with his home, and arrives at some island. After that, he will spend t day(s) on the islands. Each day, he can choose to stay and rest, or to walk to another island across the bridge. It is allowed to stay at an island for more than one day. It's also allowed to cross one bridge more than once.

Suppose that right after the t-th day DZY stands at the i-th island. Let $ans[i]$ be the number of ways for DZY to reach the i-th island after t-th day. Your task is to calculate $ans[i]$ for each i modulo 1051131.

Input

To avoid huge input, we use the following way to generate the array a. You are given the first s elements of array: $a_1, a_2, ..., a_s$. All the other elements should be calculated by formula: $a_i = (101 \cdot a_{i-s} + 10007)\ mod\ 1051131$ $(s < i \le 2^m)$.

The first line contains three integers m, t, s $(1 \le m \le 25; 1 \le t \le 10^{18}; 1 \le s \le min(2^m, 10^5))$.

The second line contains s integers $a_1, a_2, ..., a_s$ $(1 \le a_i \le 10^6)$.

Output

To avoid huge output, you only need to output xor-sum of all the answers for all i modulo 1051131 $(1 \le i \le 2^m)$, i.e. $(ans[1]\ mod\ 1051131)\ xor\ (ans[2]\ mod\ 1051131)\ xor ...\ xor\ (ans[n]\ mod\ 1051131)$.

Examples

input

```
2 1 4
1 1 1 2
```

output

```
1
```

input

```
3 5 6
389094 705719 547193 653800 947499 17024
```

output

```
556970
```

Note

In the first sample, $ans = [6, 7, 6, 6]$.

If he wants to be at island 1 after one day, he has 6 different ways:

1. home —> 1 -(stay)-> 1
2. home —> 2 —> 1
3. home —> 3 —> 1
4. home —> 3 —> 1 (note that there are two different bridges between 1 and 3)
5. home —> 4 —> 1
6. home —> 4 —> 1 (note that there are two different bridges from home to 4)

In the second sample, $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (389094, 705719, 547193, 653800, 947499, 17024, 416654, 861849)$, $ans = [235771, 712729, 433182, 745954, 139255, 935785, 620229, 644335]$.