Professor Danko teaches maths at the university. This semester N students attend his lectures in discrete mathematics and combinatorics. On his last lecture Danko spoke about ranks of permutations.

A sequence of K integers between 1 and K such that each number occurs in a sequence exactly once is called a permutation. A permutation can be rearranged into total of K! different sequences. For example a sequence (1, 2, 3) can be rearranged in 3! = 6 ways. Permutations can be ordered into a sequence of permutations by comparing first elements, than second elements, then third, and so on.

1. (1, 2, 3)
2. (1, 3, 2)
3. (2, 1, 3)
4. (2, 3, 1)
5. (3, 1, 2)
6. (3, 2, 1)

The index of a permutation in a sequence of permutations is called the rank of permutation. For example, the rank of permutation (3, 1, 2) is 5.

Danko assigned a permutation to each of his students. Each student has to calculate the rank of given permutation. Danko used this simple algorithm to generate permutations:

He picked one base permutation of K integers. A permutation for each of his student is obtained from base permutation by swapping two elements.

Danko's assistant Janko must give grades to students. Professor gave him the base permutation and a sequence of N pairs of indices that should be swapped to obtain each permutation.

For example, if the base permutation were (1, 5, 4, 2, 3), and pairs of indices were (1, 3), (2, 3) i (2, 5), then students got these three permutations: (4, 5, 1, 2, 3), (1, 4, 5, 2, 3) i (1, 3, 4, 2, 5). The ranks of permutations are 91, 17 i 9, respectively.

Write a program that will help Janko to calculate the ranks of permutations. These numbers can get very large, so output the remainder of division by 1 000 000 007 instead.

INPUT

The first line of input contains two integers K and N ($2 \le K \le 300\,000$, $1 \le N \le 100\,000$), the length of the base permutation and the number of students.

The next line contains the base permutation – a sequence of K integers between 1 and K such that each number occurs in a sequence exactly once.

The next N lines contain two integers each A and B ($1 \le A < B \le K$), indices of two elements that should be swapped to obtain each permutation.

OUTPUT

Output N integers, one per line, the rank of each permutation modulo 1 000 000 007 in the order they are presented in the input.

SCORING

Test data worth 50 points has ($1 \leq N < 10$).

SAMPLE TEST CASES

input

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

output

```
91
17
9
```

input

```
15 2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
14 15
1 15
```

output

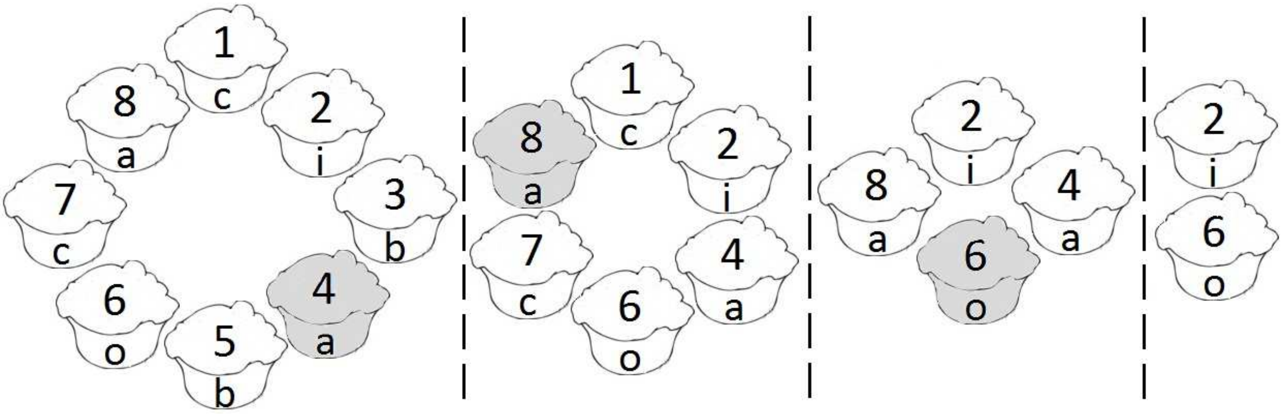
```
2
246045325
```

Darko and Marko are twins and they love to eat cookies. Their grandma Mara loves to bake cookies, but she doesn't like the fact that Darko and Marko eat them too fast.

To teach her grandsons to eat slower, Mara turned it into a game. Mara will bake N cookies and assign them with integers 1 to N. Then she will arrange them in a circle such that each cookie *i* is between to cookie *i*−1 and *i*+1 except for cookies 1 and N that are neighbors.

Mara knows a recipe for 26 different types of cookies. We will denote a cookie type with lower case english letters 'a' to 'z'.

Darko and Marko will each get one cookie every 5 minutes. Mara will say one integer out loud. Darko and Marko will search for a cookie with this number, but will eat two neighboring cookies. This procedure is repeated until one or two cookies are left on the table. Then the game ends and Mara eats the remaining cookies.



The game can be represented with a sequence of (N−1) div 2 integers that Mara said out loud. For example, the illustrations above can be represented with a sequence (4, 8, 6). Two games are considered different if their respective sequences are different.

After a few games Mara noticed that Marko and Darko often fight during the game. In fact, they fight every time when the two neighboring cookies are of different types because they can't decide which one will get which cookie.

Mara decided to count the number of ways to play a game in a way to avoid such situations.

Given a cookie type for each of N cookies, write a program that will calculate the number of ways to play a game such that Darko and Marko will not fight. This number can get very large, so output the remainder of division by 10007 instead.

INPUT

The first line contains one integer N (3 ≤ N ≤ 100), the number of cookies.

The second line contains a sequence of N lower case english letters, types of cookies in order they are arranged in a circle.

OUTPUT

Output a single integer, the total number of ways to play a game that will prevent Darko and Marko from fighting modulo 10007.

SCORING

Test data worth 30 points has ($1 \leq N < 25$).

Test data worth 60 points has ($1 \leq N < 40$).

Test data worth 80 points has ($1 \leq N < 75$).

SAMPLE TEST CASES

input

```
8
cibaboca
```

output

```
4
```

input

```
5
aabab
```

output

```
5
```

input

```
11
fffffffffff
```

output

```
388
```

Clarification of the first sample: This sample corresponds to illustrations on the previous page. Four valid sequences are (4, 8, 2), (4, 8, 6), (8, 4, 2) i (8, 4, 6).

Clarification of the second sample: Five valid sequences are (3, 1), (5, 2), (4, 4), (4, 1) i (4, 2).

Clarification of the third sample: All cookies share the same type, so Darko and Marko can't get into fight no matter what. In each step Mara can say any integer remaining on the table, so the total number of games is equal to $11 \cdot 9 \cdot 7 \cdot 5 \cdot 3 = 10395$. 388 is the remainder of division by 10007.