Sfida all'ultimo pangramma (`pangramma`)

Mojito (Monica's dog, the mascotte of OII) just finished hanging a very long banner with the help of *every other* Jack Russell Terrier living in Matera. The banner now sits in the hall of the *Pentasuglia High-School*, where the awards ceremony of the OII will soon take place. Once finished with the banner, Mojito noticed that the long list of sponsor sometimes contains **pangrams**: sentences that use every letter of the alphabet.

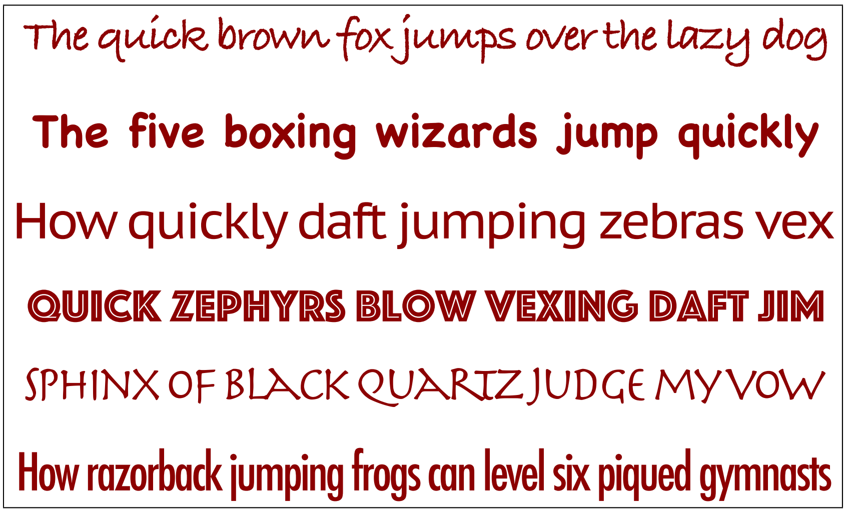

Figure 1: A selection of six different pangrams[1]

Clearly, finding long pangrams is quite easy. What's hard is to find short ones. The absolute shortest pangrams are also called **perfect pangrams**: they use each letter *exactly once*:

`Mr Jock TV quiz PhD bags few lynx`

Sadly, it's not guaranteed that a perfect pangram will appear in the banner. When this happens, Mojito will settle for **minimal pangrams** (i.e. pangrams with the lowest possible length) in the banner.

Feeling inspired from this discovery, Mojito decided to organize a game to play with his helpers: each of the **46 337** Jack Russell Terriers will have to show the others a minimal pangram from the banner. To do that, the dogs must highlight each alphabet letter in the banner in a different way than what all the other dogs did previously. Before starting this challenge, Mojito wants to know how many distinct ways there are to perform this "highlighting"... he really doesn't want to lose!

More precisely, the banner contains a string V of N characters, each character belonging to an alphabet of K symbols.[2] The symbols are represented with integers from 0 to $K-1$. A *highlighting* maps each symbol j of the alphabet to a position x_j of the string V where the symbol appears, i.e. such that $V[x_j] = j$. Two highlightings are different if they map different positions $x_j \neq x'_j$ to at least one symbol j, even if they correspond to the same pangram. A highlighting is *minimal* if it encloses a minimal pangram (i.e. if the maximum difference $x_j - x_i$ between the positions of two occurrences is the minimum possible). Mojito wants to count how many minimal highlightings exist, **modulo 46 337**: help him!

[1] Pangrams are not just Mojito's past time fun, they're actually useful in the real world. For example, pangrams are used in typography to test how different fonts render on print. Moreover, encoding a pangram is a good way to test your knowledge of Morse code. In the 80s, the Italian writer Umberto Eco was obsessed with pangrams. He found a perfect one that uses all 21 letters of the Italian alphabet:

"Tv? Quiz, BR, FLM, DC... Oh, spenga!"

[2] The banner contains letters but also: numbers, special characters, emojis, the batman symbol, ...

Implementation

> ✍ It's crucial to perform the modulo operation after each addition because the intermediate values generated might require more than 32 bit to be stored. When looking for some result modulo M, we suggest exploting the fact that $(A+B+C)$ modulo $M = (((A+B)$ modulo $M) + C)$ modulo M. The same thing is true for the multiplication. In this way, the intermediate results are always kept modulo M and can be stored in a normal integer variable.

You should submit a single file, with either a `.c` or `.cpp` extension.

> ☞ Among the attachments in this task you will find a template `pangramma.c` or `pangramma.cpp` with a sample implementation.

You will have to implement the following function:

C	`int conta(int N, int K, int* V);`
C++	`int conta(int N, int K, vector<int>& V);`

- The integer N is the length of array `V`, i.e. the total number of letters in the banner.
- The integer K is the number of different symbols in the alphabet.
- The array `V`, indexed from 0 to $N-1$, specifies the symbol contained in each position.
- The function should return the number of differet minimal highlightings modulo 46 337.

The grader will call the `conta` function and will print the returned value in the output file.

Sample grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file is formed by 2 lines:

- Line 1: two integers N and K.
- Line 2: N integers $\texttt{V}[i]$ for $i = 0 \ldots N-1$.

The output file is formed by a single line which contains the value returned by the `conta` function.

Constraints

- $1 \le K \le N \le 1\,000\,000$.
- $0 \le \texttt{V}[i] \le K-1$ for $i = 0 \ldots N-1$.
- Every symbol appears at least once in the string.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all testcases of which it is formed.

- **Subtask 1 [0 points]**: Sample testcases.
- **Subtask 2 [5 points]**: $K = 2$.
- **Subtask 3 [7 points]**: $N \le 30$.
- **Subtask 4 [14 points]**: $N \le 300$.

- **Subtask 5** [**13 points**]: $N \le 5000$.
- **Subtask 6** [**12 points**]: There is at least one *perfect pangram* in the banner.
- **Subtask 7** [**11 points**]: The minimal pangram has a length of at most 60.
- **Subtask 8** [**10 points**]: The number of minimal highlightings is at most 10^9.
- **Subtask 9** [**9 points**]: $N \le 500\,000$.
- **Subtask 10** [**10 points**]: No limits.
- **Subtask 11** [**7 points**]: ⚠*Special case:* $1\,000\,000 \le K \le N \le 5\,000\,000$.
- **Subtask 12** [**2 points**]: ⚠*Special case:* $5\,000\,000 \le K \le N \le 9\,000\,000$.

Examples

stdin	stdout
`6 3` `2 0 1 1 2 0`	`2`
`7 3` `0 1 1 2 1 1 0`	`4`
`7 3` `0 1 1 2 0 0 1`	`1`
`13 5` `2 3 0 0 3 1 3 4 3 3 0 3 2`	`10`

Explanation

In the **first sample testcase** there are two minimal pangrams, which happen to also be perfect:

$\boxed{2, 0, 1}$, 1, 2, 0

2, 0, 1, $\boxed{1, 2, 0}$

There are longer pangrams, but they should not be counted:

2, $\boxed{0, 1, 1, 2}$, 0

In total, there are **2** different ways to highlight a minimal pangram.

In the **second sample testcase** there are again two minimal pangrams, but they are not perfect pangrams. There are **4** minimal highlightings:

$\boxed{0, 1, 1, 2}$, 1, 1, 0   0, 1, 1, $\boxed{2, 1, 1, 0}$

$\boxed{0, 1, 1, 2}$, 1, 1, 0   0, 1, 1, $\boxed{2, 1, 1, 0}$

In the **third sample testcase** there is only one perfect pangram, in the middle, so there is only one minimal highlighting.

In the **fourth sample testcase** there are 2 minimal pangrams, respectively with 6 and 4 minimal highlightings, for a total of 10.