E. Keyboard Purchase

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

You have a password which you often type — a string s of length n. Every character of this string is one of the first m lowercase Latin letters.

Since you spend a lot of time typing it, you want to buy a new keyboard.

A keyboard is a permutation of the first m Latin letters. For example, if $m = 3$, then there are six possible keyboards: `abc`, `acb`, `bac`, `bca`, `cab` and `cba`.

Since you type your password with one finger, you need to spend time moving your finger from one password character to the next. The time to move from character s_i to character s_{i+1} is equal to the distance between these characters on keyboard. The total time you have to spend typing the password with a keyboard is called the *slowness* of this keyboard.

More formaly, the slowness of keyboard is equal to $\sum_{i=2}^{n} |pos_{s_{i-1}} - pos_{s_i}|$, where pos_x is position of letter x in keyboard.

For example, if s is `aacabc` and the keyboard is `bac`, then the total time of typing this password is $|pos_a - pos_a| + |pos_a - pos_c| + |pos_c - pos_a| + |pos_a - pos_b| + |pos_b - pos_c|$ = $|2 - 2| + |2 - 3| + |3 - 2| + |2 - 1| + |1 - 3| = 0 + 1 + 1 + 1 + 2 = 5$.

Before buying a new keyboard you want to know the minimum possible slowness that the keyboard can have.

Input

The first line contains two integers n and m ($1 \le n \le 10^5, 1 \le m \le 20$).

The second line contains the string s consisting of n characters. Each character is one of the first m Latin letters (lowercase).

Output

Print one integer – the minimum slowness a keyboard can have.

Examples

input
```
6 3
aacabc
```
output
```
5
```

input
```
6 4
aaaaaa
```
output
```
0
```

input
```
15 4
abacabadabacaba
```
output
```
16
```

Note

The first test case is considered in the statement.

In the second test case the slowness of any keyboard is 0.

In the third test case one of the most suitable keyboards is `bacd`.
