

B. Tyndex.Brome

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Tyndex is again well ahead of the rivals! The reaction to the release of Zoozle Chrome browser was the release of a new browser Tyndex.Brome!

The popularity of the new browser is growing daily. And the secret is not even the Tyndex.Bar installed (the Tyndex.Bar automatically fills the glass with the finest 1664 cognac after you buy Tyndex.Bottles and insert in into a USB port). It is highly popular due to the well-thought interaction with the user.

Let us take, for example, the system of automatic address correction. Have you entered `codehorses` instead of `codeforces`? The gloomy Zoozle Chrome will sadly say that the address does not exist. Tyndex.Brome at the same time will automatically find the closest address and sent you there. That's brilliant!

How does this splendid function work? That's simple! For each potential address a function of the F error is calculated by the following rules:

- for every letter c_i from the potential address c the closest position j of the letter c_i in the address (s) entered by the user is found. The absolute difference $|i - j|$ of these positions is added to F . So for every i ($1 \leq i \leq |c|$) the position j is chosen such, that $c_i = s_j$, and $|i - j|$ is minimal possible.
- if no such letter c_i exists in the address entered by the user, then the length of the potential address $|c|$ is added to F .

After the values of the error function have been calculated for all the potential addresses the most suitable one is found.

To understand the special features of the above described method better, it is recommended to realize the algorithm of calculating the F function for an address given by the user and some set of potential addresses. Good luck!

Input

The first line contains two integers n and k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^5$). They are the number of potential addresses and the length of the address entered by the user. The next line contains k lowercase Latin letters. They are the address entered by the user (s). Each next i -th ($1 \leq i \leq n$) line contains a non-empty sequence of lowercase Latin letters. They are the potential address. It is guaranteed that the total length of all the lines does not exceed $2 \cdot 10^5$.

Output

On each n line of the output file print a single number: the value of the error function when the current potential address is chosen.

Please, do not use `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use `cout` (also you may use `%I64d`).

Examples

input

```
2 10
codeforces
codeforces
codehorses
```

output

```
0
12
```

input

9 9
vkontakte
vcontacte
vkontrakte
vkollapse
vkrokodile
vtopke
vkapuste
vpechke
vk
vcodeforcese

output

18
14
36
47
14
29
30
0
84