String Modification

Roy was given a string s containing only uppercase English letters. He can do any number of modifications on s. The allowed modifications are:

1. He can add underscore ('`_`') character in anywhere inside the string.
2. He can delete any existing character of the string.
3. He can swap any two characters of the string.

Every character in the resulting string has a value equal to its ASCII value.

After doing the modifications the string needs to have the following properties:

1. The length of the string should be equal to n.
2. There should be at least k characters of higher value between two equal letters (Note that, underscore is not a letter).

Calculate how many different strings Roy can achieve **modulo** $1000003\ (10^6 + 3)$.

Note: *In the increasing order of ASCII value, we can arrange the alphabet in the following way,* `A` $<$ `B` $<$ `C` $<$ `D` $< \cdots <$ `X` $<$ `Y` $<$ `Z` $<$ `_`

Input Format

The first line contains two space separated integers $n\ (1 \le n \le 10^9)$ and $k\ (0 \le k \le 10^9)$. The second line contains string s containing only uppercase English letters $(1 \le |s| \le 2500)$.

Output Format

Print the number of different strings Roy can achieve modulo $1000003\ (10^6 + 3)$.

Sample Input #1

```
3 1
LBB
```

Sample Output #1

```
15
```

Sample Input #2

```
5 2
PPPP
```

Sample Output #2

```
9
```

Sample Input #3

```
8 7
DQ
```

Sample Output #3

```
73
```

Sample Input #4

```
1078 223
RMXQYQPKSSBJCAFWPXZ
```

Sample Output #4

```
451838
```

Explanation

In the first test case, the 15 valid strings are

```
BLB
BL_
B_B
B_L
B__
LB_
L_B
L__
_BL
_B_
_LB
_L_
__B
__L
___
```