I. Increasing Sequence

Problem ID: increasing

Given an increasing sequence $A_1, A_2, \cdots, A_N$ and an integer S, calculate the number of triplets (i, j, k) satisfying

- $1 \leq i < j < k \leq N$
- $A_j - A_i > S$ and $A_k - A_j > S$

Input

The first line contains two integers N and S, representing the length of the sequence and the parameter S described in the problem.

The second line contains n integers $A_1, A_2, \cdots, A_N$.

Output

Output an integer as the answer.

Subtasks

For all subtasks: $0 \leq A_i, S \leq 10^9, 3 \leq N \leq 3 \times 10^6$.

1. 12 points ($\star$) : $N \leq 500$
2. 12 points ($\star\star$) : $N \leq 5000$
3. 6 points ($\star\star\star$) : $N \leq 2 \times 10^5$

Hints

- Any algorithm with complexity $O(N^3)$ should be enough for subtask 1.
- Any algorithm with complexity $O(N^2)$ should be enough for subtask 2.
- Any algorithm with complexity $O(N)$ or $O(N \log N)$ should be enough for subtask 3.

Sample Input 1	Sample Output 1
5 1 1 2 3 4 5	1

The only triplet is $(1, 3, 5)$.

Sample Input 2	Sample Output 2
5 0 0 0 1 2 2	4

All triplets satisfying the conditions are $(1, 3, 4)$,$(1, 3, 5)$,$(2, 3, 4)$,$(2, 3, 5)$.

Sample Input 3	Sample Output 3
6 3 5 5 6 6 7 7	0

No triplet satisfies the conditions.

II. Prefix and Suffix

Problem ID: prefixandsuffix

Given a string S consisting of lower-case letters and an integer K, you are asked to calculate the number of strings T consisting of only lower-case letters with length not exceeding K such that S is both a prefix and a suffix of T. Please print the answer modulo $10^9 + 7$.

Input

The first line contains two integers N and K, representing the length of the string S and the parameter K described in the problem.

It is guaranteed that S comprises only lower-case letters.

The second line contains the string S.

Output

Output the answer modulo $10^9 + 7$.

Subtasks

For all subtasks: $1 \leq N \leq 10^6, N \leq K \leq 10^9$.

1. 10 points ($\star\star$) : $N \leq 5000, K \leq 5 \times 10^6$
2. 18 points ($\star\star\star$) : $N \leq 10^6, K \leq 5 \times 10^6$
3. 2 points ($\star\star\star\star$) : $N \leq 10^6, K \leq 10^9$

Hints

- Any algorithm with complexity $O(N^2)$ should be enough for subtask 1.
- Any algorithm with complexity $O(N + K)$ should be enough for subtask 2.
- Any algorithm with complexity $O(N)$ should be enough for subtask 3.
- You may consider the following cases separately:
  1. $2 \cdot |S| > |T|$.
  2. $2 \cdot |S| \leq |T|$.
- You may want to use Fermat's Little Theorem to solve the last subtask:

$$\text{for all prime number } p,\ x^{p-2} \cdot x \equiv 1 \pmod p, \forall x \in \mathbb{Z}, \gcd(x, p) = 1$$

Sample Input 1	Sample Output 1
5 9 ababa	3

There are 3 strings satisfying the condition:

- ababa
- abababa
- ababababa

Sample Input 2	Sample Output 2
2 5 ab	28

There are 28 strings satisfying the condition:

- ab
- abab
- abaab
- abbab
- abcab
- abdab

- `abeab`
- `abfab`
- `abgab`

...

- `abzab`

Sample Input 3	Sample Output 3
5 9 aabab	1

The only valid string is `aabab`.

Sample Input 4	Sample Output 4
4 8 cabc	3

There are 3 strings satisfying the condition:

- `cabc`
- `cabcabc`
- `cabccabc`

Sample Input 5	Sample Output 5
6 30 aabaab	865159427

Sample Input 6	Sample Output 6
7 100 abacaba	246659705

III. Longest Common Subsequence

Problem ID: lcs

Given two strings X and Y that consist of characters D, S and A. Also, four integers C_D, C_S, C_A, M are specified, representing respectively the cost of each character and the money you have initially (this will become clear later).

You can perform the following operation arbitrary number of times (as long as you have enough money): insert a character $x \in \{D, S, A\}$ to an arbitrary position of one of the two strings and pay C_x dollars. Find out the maximum length of the longest common subsequence of the two strings you can obtain by performing a sequence of operations, assuming you have M dollars initially.

Input

The first two lines contain a string each, representing X and Y. It's guaranteed that both strings consist of only characters D, S and A.

The next line consists of three positive integers C_D, C_S and C_A.

The last line contains a non-negative integer M.

Output

Output a line consisting of the maximum length of the longest common subsequence you can obtain by a sequence of operations.

Subtasks

For all subtasks: $1 \le |X|, |Y| \le 300, 0 \le M \le 300, 1 \le C_D, C_S, C_A \le 300$.

1. 15 points ($\star$) : $M = 0, 1 \le |X|, |Y| \le 300$
2. 5 points ($\star\star\star$) : $1 \le |X|, |Y|, M \le 50$
3. 5 points ($\star\star\star\star$) : $1 \le |X|, |Y|, M \le 100$
4. 5 points ($\star\star\star\star\star$) : $1 \le |X|, |Y|, M \le 300$

Sample Input 1	Sample Output 1
DSADSADSA ASDASDASD 10 10 10 0	5

No operations can be done in this case. One of the longest common subsequences is DADAD.

Sample Input 2	Sample Output 2
DSA DSA 1 2 3 4	5

By inserting four D into the strings we get two strings DDSAD and DDSAD, and the longest common subsequence of them is DDSAD.

Sample Input 3	Sample Output 3
DASSD DASAA 10 1 10 8	7

Sample Input 4

```
DDASS
SSADD
8 3 1
4
```

Sample Output 4

```
5
```

Sample Input 5

```
SDDDA
ASDAS
4 9 1
1
```

Sample Output 5

```
4
```

The first string becomes `ASDDDA` if we spend one dollar and insert an `A` into the first position of the first string. `ASDA` is the longest common subsequence of the two strings after the operation.

Sample Input 6

```
AASAA
AAASD
1 6 10
5
```

Sample Output 6

```
6
```

Append three `D` to the end of the first string and two `D` to the end of the second string. `AASDDD` is their longest common subsequence.

Sample Input 7

```
SSSS
AAAA
3 1 2
12
```

Sample Output 7

```
8
```

Insert four `A` in the beginning of the first string and four `S` in the end of the second string. Both strings become `AAAASSSS` and the longest common subsequence of them has length 8.

IV. Shortest Path

Problem ID: shortestpath

Given a weighted directed graph without self-loops and multiple edges of N vertices and M edges (vertices are numbered from 1 to N). Find

1. The length of the shortest path from vertex 1 to vertex 2.
2. The number of shortest paths from vertex 1 to vertex 2. Two shortest paths $a_1 = 1, a_2, \ldots, a_p = 2$ and $b_1 = 1, b_2, \ldots, b_q = 2$ (where a_i and b_i corresponds to the i-th vertex on the two shortest paths, respectively) are considered different if $p \neq q$ or $p = q$ and there exists an index $1 \leq i \leq p$ such that $a_i \neq b_i$. As the answer might be large, print it modulo $10^9 + 7$.
3. The number of *critical* edges. An edge e is critical if the length of the shortest path from vertex 1 to vertex 2 increases (or vertex 2 becomes unreachable from vertex 1) if we remove e from the graph.

It's guaranteed that vertex 2 is reachable from vertex 1 in the given graph.

Input

The first line of the input contains two integers N and M, corresponding to the number of vertices and the number of edges respectively.

M lines follow, the i-th of which consists of three positive integers u_i, v_i and w_i, representing an directed edge from u_i to v_i of weight w_i.

Output

Output three lines. Print the answer to the i-th question on the i-th line (50% for the first question; 30% for the second question and 20% for the third).

Note that no points will be awarded if the output contains an incorrect number of lines.

Subtasks

For all subtasks: $1 \leq N \leq 10^5, 0 \leq M \leq \min(2 \times 10^5, N \times (N-1)), 1 \leq u_i, v_i \leq N, 1 \leq w_i \leq 2 \times 10^9$.

1. 6 points ($\star\star$) : $1 \leq N \leq 10$
2. 7 points ($\star\star$) : $1 \leq N \leq 100$
3. 8 points ($\star\star\star$) : $1 \leq N \leq 1000$
4. 9 points ($\star\star\star\star\star$) : $1 \leq N \leq 10^5$

Sample Input 1	**Sample Output 1**
5 5 1 3 1 3 4 3 1 5 2 5 4 2 4 2 4	8 2 1

The picture below illustrates the graph of sample input 1.

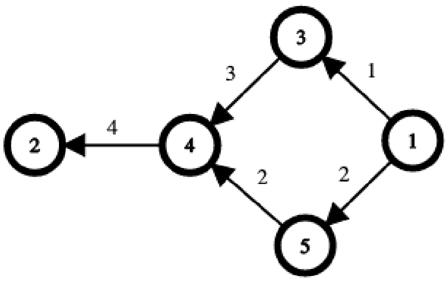

Figure IV.1: In sample input 1, the length of the shortest paths from 1 to 2 is 8. There are two shortest paths from 1 to 2: $1 \to 3 \to 4 \to 2$ and $1 \to 5 \to 4 \to 2$. The last edge is the only critical edge.

Sample Input 2	**Sample Output 2**
4 6 1 2 7 2 1 1 1 3 2 4 2 3 3 4 2 4 3 3	7 2 0

Sample Input 3	**Sample Output 3**
4 6 1 2 9 2 1 1 1 3 2 4 2 3 3 4 2 4 3 3	7 1 3