# Problem D. Viral bacterium

Input file: bacteria.in
Output file: bacteria.out

Almaty was attacked by a dangerous bacterium! An experienced fighter — Azizkhan was invited to deal with it. Bacterium consists of several cells, ordered in a row. The scientists know that there are only 26 different type of cells: each cell is represented by a capital latin letter. More formally, each bacterium is represented by a string of length $N$, which contains only capital latin letters.

To destroy bacterium Azizkhan can shoot at some position in the bacterium (between two neighbour cells, or one of the ends) by the cell of any type. If at some moment, there are at least $M$ consecutive cells of the same type, all these cells disappear. After that happens, cells located to the sides of disappeared group **concatenate**. What is the minimal required number of shots Azizkhan needs to take in order to kill the bacterium, i.e. disappear all cells.

## Input

In the first line of input there are two positive integer numbers $N$ and $M$ — size of the bacterium and the minimal number of consecutive cells of the same type which disappear.

Second line contains string $S$ — representation of bacterium, which consists of $N$ capital latin letters. It is guaranteed that the string **does not contain** $M$ consecutive same symbols.

## Output

Print out a single number — answer to the problem.

## Examples

| bacteria.in | bacteria.out |
|---|---|
| 4 2<br>ABAB | 2 |

## Scoring

This problem consists of four subtasks:

1. $1 \le N \le 20$, $M = 2$, there is only two type of cells: 'A','B'. This subtask worths 6 points.
2. $1 \le N \le 300, M = 2$. This subtask worths 22 points.
3. $1 \le N \le 300, 2 \le M \le 3$. This subtask worths 33 points.
4. $1 \le N \le 500, 2 \le M \le 20$. This subtask worths 39 points.

Each subtask will be scored only if the solution successfully passes all of the previous subtasks.

# Problem E. Russian checkers

The board for the game called "Russian checkers"is similar to the chessboard. It is a 8x8 square, however, unlike the chessboard, only the black squares are used. They can be numbered as it is shown in the picture 1.

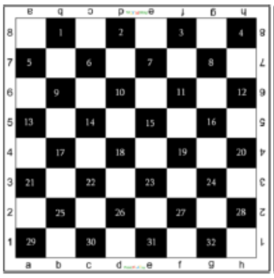
рис. 1

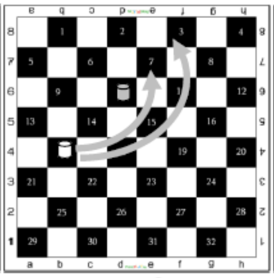
рис. 2

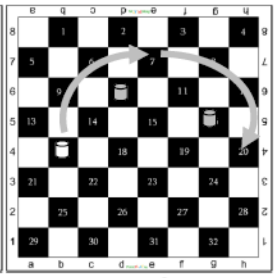
рис. 3

We will consider one of the possible final states of the game, where three kings of one colour play against one king of another colour. Players alternate turns. In one turn, players can move one of the kings of their colour to any distance along the chosen diagonal. The winner is the one who managed to destroy or to block all of the opponent's kings. There are two types of king's moves:

- Regular move - is a movement along the diagonal. It is allowed when it is impossible to capture one of the opponent's kings.
- Capturing move - is a movement over the opponent's king. In such case, the king must capture the opponent's king if they are located on the same diagonal and there is at least one empty square behind the opponent's figure. The king which makes the capturing move can be placed on one of those empty squares, jumping over the opponent's king. For example, as it is shown on the picture 2, the white king placed on 17 square can make a capturing move to the square 7 or 3. This move can be done in any direction.

It is not allowed to jump over the same opponent's king more than once. **If there are some opponent's kings with the empty squares behind them on any of the diagonals that cross our current one, we must capture all of them.** (It's shown on picture 3). Captured kings are taken from the board only after the capturing one finishes its move. If several kings can make a capturing move, or it can be done in several directions, the choice is on the player.

The configuration of the game with 3 white and 1 black kings is given. The player with white kings moves first. It is guaranteed that the player with white figures can win in the initial configuration. Write a program that will play with the white figures and will win the game in no more than fixed number of moves (see scoring).

In case your program makes an invalid move, the execution of the program will be stopped and the "Wrong Answer"report will occur.

## Example

White kings are located on squares 21, 14 and 26. Black king is on the position 26. White figures can win in two moves.

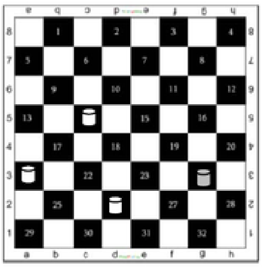
рис. 4

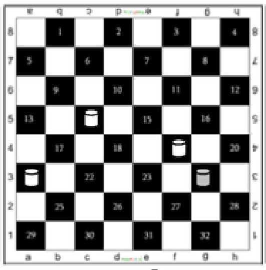
рис. 5

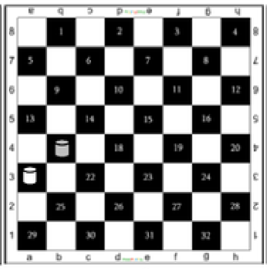
рис. 6

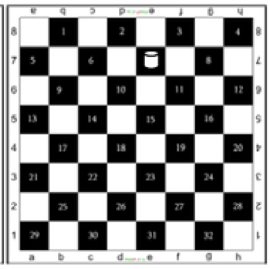
рис. 7

Explanation of the example: the initial configuration is shown in the picture 4. Using the optimal strategy, whites make the move 26-19 (picture 5). Blacks are forced to capture the kings on 19 and 14. (picture 6). After that, white king on 21 captures the black one on 17 (picture 7) and whites win. Besides square 7, the white king could also make a capturing move to positions 14, 10 or 3.

## The interaction protocol

You have to implement two following procedures:

$initialize(positions)$ - this procedure will be called by grader at the beginning of the program execution exactly once.

$positions$: An array of integers - the initial positions of the kings, $positions_0, positions_1, positions_2$ denote the initial locations of the white kings, $positions_3$ is a location of the black king.

$whitemove(black\_position, move)$ - this procedure should execute white king's move, it will be called by grader several times.

$black\_position$: location of the black king after its move. It is guaranteed that the move was valid.

$move$: array of integers of length 2. $move_0$ should be the location of the white king which makes the move, and $move_1$ should be the location of the destination of this king after the move.

## Scoring

This problem consists of 5 subtasks.

1. Victory in 1 move: white king captures the black one in one move. This subtask worths 7 points.
2. Victory in 3 moves: whites make 2 moves, blacks make 1 move, it is guaranteed that there is a strategy where whites lose no kings. This subtask worths 10 points.
3. Victory in 3 moves: Whites make 2 moves and blacks make 1 move. This subtask worths 17 points.
4. Victory in 7 moves: whites make 4 moves, blacks make 3 move. This subtask worths 29 points.
5. Full solution in no more than 15 moves of whites. This subtask worths 47 moves.

# Problem F. Travel convenience

Input file: `hamilton-cycle.in`
Output file: `hamilton-cycle.out`

There are $N(3 \leq N \leq 5000)$ towns in the country called Oz. These towns are numbered from 1 to $N$. Each town has its own beauty - a positive integer number. The airplanes of the `OzAir` airline, which is the most successful (and the only) airline of this country, can travel between two towns only if the difference of values of their beauty does not exceed the fixed value $D(1 \leq D \leq 10^9)$. It means that the town $v$ can be reached directly from the town $u$ if $|a_v - a_u| \leq D$, where $a_v$ is a beauty of the town $v$.

The Wizard of Oz wants to make Oz country convenient for travelling. He has his own conception of travel convenience, so the country is considered as convenient for travelling if it is possible to travel through all of the towns, return to the initial town in the end of the route and visit each town besides the initial one only once. More formally, the country is convenient for travelling if $m > 2$ and there is a permutation $p_1, p_2, ..., p_m$, so that for each $i(1 \leq i < m)$ there is a direct flight from the town $p_i$ to the town $p_{i+1}$, also there should be a direct flight between the towns $p_m$ and $p_1$.

The Wizard of Oz may destroy the towns of Oz. He wants to know the number of ways to destroy some towns, so that the Oz country becomes convenient for travelling. This number can be large, so calculate it modulo $1\,000\,000\,007(10^9 + 7)$.

## Input

First line of the input file contains two positive numbers $N(3 \leq N \leq 5000)$ и $D(1 \leq D \leq 10^9)$ — the number of towns and the maximum acceptable difference of the beauties of two towns. Next line contains $N$ positive numbers $a_i(1 \leq a_i \leq 10^9)$ — the values of beauty of the towns.

## Output

Print one number - the answer for the problem modulo $1000000007(10^9 + 7)$.

## Examples

| hamilton-cycle.in | hamilton-cycle.out |
|---|---|
| 3 3<br>1 2 3 | 1 |
| 4 4<br>1 1 1 1 | 5 |

## Scoring

This problem consists of three subtasks:

1. $3 \leq N \leq 15$. This subtask worths 9 points.
2. $3 \leq N \leq 500$. This subtask worths 31 points.
3. $3 \leq N \leq 5000$. This subtask worths 60 points.

Each subtask will be scored only if the solution successfully passes all of the previous subtasks.