Hong Kong Olympiad in Informatics 2015/16
Senior Group

Task Overview

ID	Task	Subtasks	Max Score
S161	Military Training	$19 + 14 + 17 + 28 + 22$	100
S162	Robos' Feast	$20 + 35 + 45$ (Partial score available)	100
S163	Arithmetic Sequence	$13 + 15 + 25 + 21 + 26$	100
S164	Alice's Meal	$11 + 9 + 27 + 15 + 38$	100

For all tasks:
CPU time limit: 1 second
Memory limit: 256 MB

Notice:

Unless otherwise specified, inputs and outputs shall follow the format below:
- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- No trailing space(s) in each line.
- No empty lines, except that the input and output should end with the endline character.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially slower performance.

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long int`.

C/C++ programmers should use `"%lld"` for 64-bit integers I/O.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

Military Training

Problem

In Byteland, the army consists solely of robots. One good thing about a robotic army is that the soldiers follow orders with absolute obedience.

Robo is a new type of robots, and they are undergoing military training. The basic training improves their ability to perform repetitive movement. The training takes place in a $N \times N$ grid. The top-left corner of the grid is $(1, 1)$ and the bottom-right corner of the grid is (N, N). At time 0, Robo is located at square (r_0, c_0) in the grid and it receives some commands, given as a string of length K. Each character of the string is either `U`, `D`, `L`, or `R`, representing the four directions — up, down, left, and right.

Each unit of time, Robo does the same thing. First, it reads the first character in the string of commands. If the character is `U`, it moves one step up; if the character is `D`, it moves one step down; similar for `L` and `R`. Then, it checks whether its current position is out of bounds. If its current location is outside the $N \times N$ grid, the training ends immediately. Otherwise, it moves the first character of the string of commands to the back of the string.

Recently, a new element is added to the basic training to improve Robos' combat ability. At time 0, a dummy robot is placed at square (r_1, c_1) in the grid and it stays there until the training ends. Whenever Robo meets the dummy robot (that is, when they are in the same square), it attacks the dummy robot once. Note that if you place the dummy robot at (r_0, c_0), the Robo will not attack it immediately. Instead, the Robo will attack the dummy robot only when it revisits (r_0, c_0) after some time.

The general of Byteland army wants to know how effective the new training is. Your task is to find out the maximum number of times Robo would attack the dummy robot, if the dummy robot is placed at an optimal position.

Input

The first line contains two integers, N and K.
The second line contains two integers, r_0 and c_0. $(1 \leq r_0, c_0 \leq N)$
The third line contains a string of length K, the command Robo receives at time 0. The string contains only `U`, `D`, `L`, and `R`. It is guaranteed that Robo's position at time K will not be (r_0, c_0).

Output

Output one integer, the maximum number of times the Robo would attack the dummy robot.

Sample test

Input	Output
3 2 1 1 LU	0

Input	Output
3 5 1 1 DDURL	3

Input	Output
2 6 1 1 RLDUUU	2

Explanation

In sample test 1, Robo is already outside the grid at time 1, so it would not carry out the attack command.
In sample test 2, the path of Robo is $\rightarrow (2,1) \rightarrow (3,1) \rightarrow (2,1) \rightarrow (2,2) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (4,1)$.
The dummy robot should be placed at $(2, 1)$.
In sample test 3, the path of Robo is $\rightarrow (1,2) \rightarrow (1,1) \rightarrow (2,1) \rightarrow (1,1) \rightarrow (0,1)$.
The dummy robot should be placed at $(1, 1)$.

Subtasks

Subtask 1 (19 points)
$N = 5$
$K = 3$
$r_0 = c_0 = 3$
The first command is `U`

Subtask 2 (14 points)
$1 \leq N \leq 1000$
$1 \leq K \leq 20$
Command consists only of `L` and `R`

Subtask 3 (17 points)
$1 \leq N \leq 1000$
$1 \leq K \leq 20$

Subtask 4 (28 points)
$1 \leq N \leq 10^9$
$1 \leq K \leq 50$

Subtask 5 (22 points)
$1 \leq N \leq 10^9$
$1 \leq K \leq 2000$

Robos' Feast

Problem

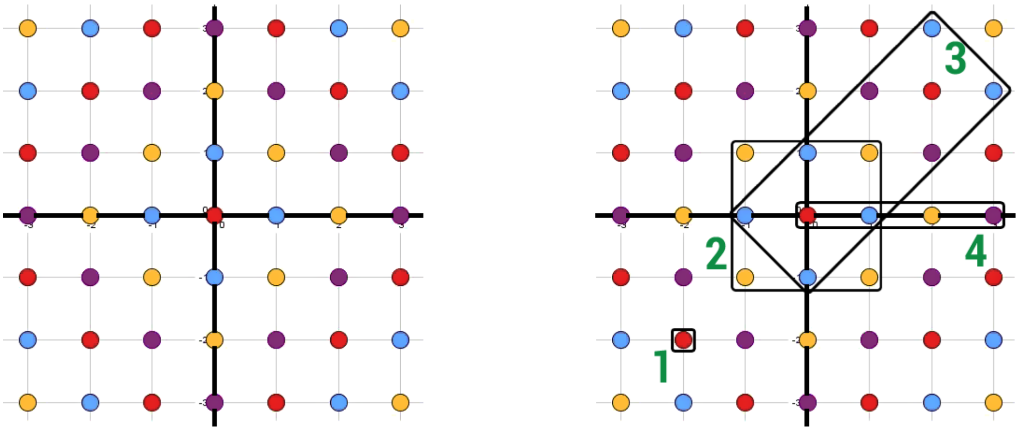
You may wonder what the diet of an intelligent robot is like. In Byteland, Robo robots have a simple diet – they only consume bottled oil. Normally, only one oil flavor is available. On the special Annual Robos' Festival, however, four different oil flavors are available. The bottles containing different oil flavors have different colors: red, blue, orange, and purple.

Every year, during Annual Robos' Festival, a feast is held. To prepare for the feast, robot helpers have to place the bottled oil on the ground to form a pattern.

Let's treat the ground as a Cartesian coordinate plane. Recall that the Manhattan distance between two points (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$. For each point (x, y) with integer coordinates, consider the remainder of the Manhattan distance between $(0, 0)$ and (x, y) when divided by 4.

If the remainder is 0, a red (R) oil bottle is placed at (x, y).
If the remainder is 1, a blue (B) oil bottle is placed at (x, y).
If the remainder is 2, an orange (O) oil bottle is placed at (x, y).
If the remainder is 3, a purple (P) oil bottle is placed at (x, y).

Here is an illustration of the pattern near the origin $(0, 0)$.
A color version is printed separately on the rough work sheet.



Different flavors of bottled oil have different tastiness. The tastiness of the four oil flavors, red, blue, orange and purple, are described by four integers T_R, T_B, T_O, T_P respectively.

This year, N Robos join the feast. They queue up and take a number of oil bottles, in a specified manner. In a Robo's turn, it performs one of the following actions:

1. Specify a rectangle with sides parallel to the axes and take all oil bottles in the rectangle (including those on the boundary).
2. Specify a rectangle with sides making an angle of 45 degrees with the axes and take all oil bottles in the rectangle (including those on the boundary).

Afterwards, new oil bottles are placed to restore the pattern.

Your task is to find the sum of tastiness of the oil bottles taken by each Robo.

Input

The first line contains four integers, T_R, T_B, T_O and T_P.
The second line contains an integer N.
The i^{th} line of the following N lines describes the i^{th} Robo's action. Each line contains five integers, C, x_1, y_1, x_2, y_2. If $C = 1$, the robot performs action 1. Otherwise, $C = 2$ and the robot performs action 2. (x_1, y_1) and (x_2, y_2) are the opposite corners of the rectangle that the robot specifies. Note that a given rectangle may have zero area.

Output

Output N lines. The i^{th} line should be the sum of tastiness of the oil bottles taken by the i^{th} Robo.

Sample test

Input	Output	Partial Score Output	Partial Score Output
`1 2 4 8`	`1`	`1`	(any integer)
`4`	`25`	`25`	(any integer)
`1 -2 -2 -2 -2`	`34`	(any integer)	`34`
`1 -1 1 1 -1`	`15`	`15`	(any integer)
`2 0 -1 2 3`			
`1 3 0 0 0`			

Explanation

Robo 1: 1 Red. Sum = $1 \times 1 = 1$
Robo 2: 1 Red, 4 Blues and 4 Oranges. Sum = $1 \times 1 + 4 \times 2 + 4 \times 4 = 25$
Robo 3: 2 Reds, 6 Blues, 1 Orange, 2 Purples. Sum = $2 \times 1 + 6 \times 2 + 1 \times 4 + 2 \times 8 = 34$
Robo 4: 1 Red, 1 Blue, 1 Orange, 1 Purple. Sum = $1 \times 1 + 1 \times 2 + 1 \times 4 + 1 \times 8 = 15$

Subtasks

Subtask 1 (20 points)
$0 \le T_R = T_B = T_O = T_P \le 100$ (The four flavors of bottled oil have the same tastiness)
$1 \le N \le 10000$
$-10^8 \le x_1, y_1, x_2, y_2 \le 10^8$

Subtask 2 (35 points)
$0 \le T_R, T_B, T_O, T_P \le 100$
$1 \le N \le 100$
$-100 \le x_1, y_1, x_2, y_2 \le 100$

Subtask 3 (45 points)
$0 \le T_R, T_B, T_O, T_P \le 100$
$1 \le N \le 10000$
$-10^8 \le x_1, y_1, x_2, y_2 \le 10^8$

Scoring

Within a subtask, if your program gives correct output for at least one type of action for each test case, you will score 80% of that subtask (16, 28 and 36 points respectively). Note that your program should still output N lines even if it is intended to solve for only one type of action.

Arithmetic Sequence

Problem

Dr. Jones is a professor in Byteland Academy. Recently, he taught his class what is arithmetic sequence. A sequence is an arithmetic sequence if the difference between two consecutive terms is constant. For example, 1, 3, 5 and 7, 5, 3 are arithmetic sequences.

Today, Dr. Jones is very tired during lesson time. Therefore, he decides to ask students questions instead of teaching anything so that the lesson is more relaxing to him. Dr. Jones takes out R cards with $1, 2, \ldots, R$ written on them respectively. Then, he asks the best student in the class, Alice, to pick N cards out of the R cards and arrange them in a line from left to right, in a special way.

Now, consider the N integers written on the N cards in the line from left to right. The N integers form a sequence. Let's call the sequence S so S_i is the number written on the i^{th} card from the left. Dr. Jones requests that Alice should arrange the cards such that sequence formed by the N integers does **NOT** contain any arithmetic **sub-sequence** of length 3. Which means, there should not exist (i, j, k) where $1 \leq i < j < k \leq N$ such that $S_j - S_i = S_k - S_j$.

For example, let $R = 7$ and $N = 5$. The sequences 1, 3, 7, 6, 2 and 4, 6, 1, 2, 5 are examples of valid answers. However, 1, 3, 7, 5, 6 is not allowed because there exists a sub-sequence 1, 3, 5 which is an arithmetic sequence.

Please help Alice to choose and arrange the cards.

Input

The first and only line contains two integers, R and N.
Inputs in the test cases guarantee that at least one solution exists.

Output

Output N space-separated integers in one line, representing an arrangement of cards satisfying Dr. Jones' request. If there are more than one arrangement, output any one of them. There should not be any space after the last number.

Sample test

Input	Output
2000000 5	1 3 7 6 2

Input	Output
4 4	3 4 1 2

Subtasks

Subtask 1 (13 points)
$1 \leq N \leq 10$
$R = 2000000$

Subtask 2 (15 points)
$1 \leq N \leq 10$
$R = N$

Subtask 3 (25 points)
$1 \leq N \leq 250$
$R = 4000$

Subtask 4 (21 points)
$1 \leq N \leq 10000$
$R = 2000000$

Subtask 5 (26 points)
$1 \leq N \leq 10000$
$R = N$

Alice’s Meal

Problem

After attending Dr. Jones’ class, Alice feels very tired and hungry. She goes to a restaurant to have a meal. There are N dishes on the menu. The dishes on the menu are numbered from 1 to N from top to bottom. After reading the whole menu, Alice knows how delicious the dishes are and gives each dish a delicious level. Dish i has a delicious level of D_i.

Since Alice is in a hurry, she has only K minutes to eat, and each minute she can finish exactly one dish. Moreover, the later she eat, the happier she would become, even for the same dish. This is because she is so hungry that she enjoy the first few dishes less. Specifically, if Dish i is eaten during the t^{th} minute, Alice will gain $t \times D_i$ happiness from the dish. The total happiness Alice can gain from the meal is the sum of the happiness from each dish.

Alice is too lazy to order the dishes one by one, so she wants to pick exactly K consecutive dishes in the menu, and eat them according to the menu order from top to bottom. At the same time, she wants to gain as much happiness as possible. For example, there are 5 dishes in the menu, and their delicious level $D_1, \ldots, D_5$ are 2, 4, 1, 5 and 3. If Alice has 3 minutes to eat, she should choose Dish 2 to 4, which can give her $1 \times 4 + 2 \times 1 + 3 \times 5 = 21$ happiness.

As a friend of Alice, help her find the maximum total happiness she can gain from the meal.

Input

The first line contains two integers, N and K.
The second line contains N integers. The i^{th} integer is the happiness value of Dish i: D_i.

Output

Output one integer, the maximum possible value of the total happiness.

Sample test

Input	Output
5 3 2 4 1 5 3	21

Input	Output
6 2 3 1 1 1 1 1	5

Subtasks

Subtask 1 (11 points)
$1 \leq N \leq 2000$
$K = 1$ which means Alice can only eat 1 dish
$0 \leq D_i \leq 1000$

Subtask 2 (9 points)
$1 \leq N \leq 2000$
$K = N$
$0 \leq D_i \leq 1000$

Subtask 3 (27 points)
$1 \leq N \leq 2000$
$1 \leq K \leq N$
$0 \leq D_i \leq 1000$

Subtask 4 (15 points)
$1 \leq N \leq 100000$
$1 \leq K \leq N$
$0 \leq D_1 \leq D_2 \leq ... \leq D_{N-1} \leq D_N \leq 1000$

Subtask 5 (38 points)
$1 \leq N \leq 100000$
$1 \leq K \leq N$
$0 \leq D_i \leq 1000$