Hong Kong Olympiad in Informatics 2014
Junior Group

Task Overview

Task	CPU time limit	Score
Enumeration	1 second	50
Phone Menu	1 second	100
Magic Stones	1 second	100
Power Socket	1 second	100
Fair Santa Claus	1 second	100

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.

C/C++ programmers should use `"%I64d"` for 64-bit integers I/O.

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

Enumeration

Time Limit: 1 second

Problem

Write a program to read an integer N and output N^2 integers in the format specified below.

Input

The input contains an integer N ($1 \leq N \leq 10$).

Output

The output consists of N lines.
The first integer on the i-th line is i^2, followed by $N-1$ integers, each of them is i greater than the integer on its left.

Sample test

Input	Output
4	1 2 3 4 4 6 8 10 9 12 15 18 16 20 24 28

Phone Menu

Time Limit: 1 second

Problem

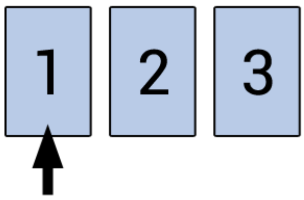
As you may have heard from the Senior Group contestants, Dr. Jones has got a new smartphone called the hkoiPhone. He then downloaded a lot of games and apps. You are interested in one of the games called 'Tower and Dragons' so you want to borrow his new phone. There are total N pages of apps. When you open the app menu, it displays the first page. Tower and Dragons is located on the K-th page so you want to go to that page to launch the game.

Sadly, the hkoiPhone is very outdated. The pages are not all loaded immediately. At time $t = 0$ seconds, M pages are already loaded. For every T_m seconds afterwards, another M pages will be loaded (all pages will be loaded if there are fewer than M pages remaining), and so on. You can slide the screen to change pages, each taking T_s seconds. You can either move between neighboring, loaded pages or between the first page and the last page loaded at the time you start sliding. Of course, you can stay at the current page at any time.

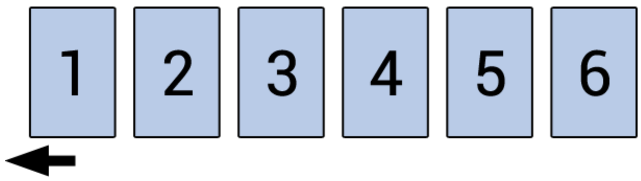
Find the minimum time required to reach the K-th page.

Here is an example. Suppose there are 8 pages and the game is on the 6th page ($N = 8, K = 6$). Also, 3 pages are loaded in 10 second intervals and it takes 1 second to slide between pages. ($M = 3, T_m = 10, T_s = 1$). The below illustrates one of the ways to achieve the minimum required time (11 seconds).

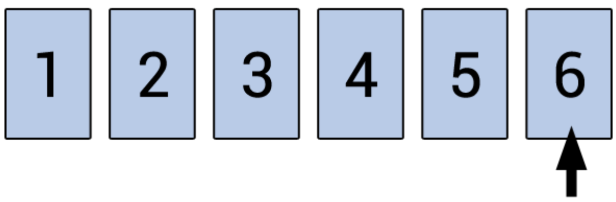
At $t = 0$ seconds, 3 pages are already loaded. You should wait.



At $t = 10$ seconds, 3 more pages are loaded. At the same time, start sliding from the first page to the last (6th) page.



At $t = 11$ seconds, you reach the 6th page, which the game is located on.



Input

The input contains 5 integers in a single line, N, M, K, T_m and T_s.

Output

Output one integer, the minimum time to reach the K-th page. You can assume that the answer must be less than 2^{31}.

Sample test

Input	Output
8 3 6 10 1	11

Input	Output
15 4 7 10 1	12

Constraints

In test cases worth 50% of the total points, $1 \le M, K \le N \le 100$.

In all test cases, $1 \le T_m, T_s \le 10000$. $1 \le M, K \le N \le 10^9$.

Besides, in test cases worth 30% of the total points, N is a multiple of M.

Magic Stone

Time Limit: 1 second

Problem

'Tower of Dragons' is a popular puzzle game co-designed by developers from Japan and Hong Kong. Inside the game, the player has to collect monsters and fight against enemies. One way to collect powerful monsters is to use magic stones.

One needs to spend real money to buy magic stones. The game company does not want to spoil the fun of players who do not spend money. Therefore, the system sometimes gifts free magic stones to players. A player can spend magic stones whenever he or she wants. He or she can also choose which magic stones, bought, gifted or a mix of those, are to be spent.

Dr. Jones spends a lot of money on magic stones. But he is a wise spender, he knows when the system will gift magic stones. He has also planned when he will buy magic stones and when he will use them. He has written down N events in his log book. The events are sorted by time. Each event is one of the following types.

Type $T_i = 1$: Buy A_i magic stones.
Type $T_i = 2$: The system gives you A_i magic stones.
Type $T_i = 3$: Dr. Jones uses A_i magic stones.
Type $T_i = 4$: Query: Among the magic stones Dr. Jones currently has, at most how many of them were bought? And at least how many of them were bought?

It is guaranteed that Dr. Jones always has enough magic stones to spend. Now you are asked to answer the queries.

Input

The first line contains an integer N.
The $(i + 1)$-th line describes the i-th event. The first integer is T_i, denoting the type of event. If $T_i = 1$, 2 or 3, the line contains a second integer A_i.

Output

For each Query event, output a line containing two integers. The first is the maximum number of bought magic stones and the second is the least number of that. Output in the order as given in the input.

Sample test

Input	Output
6 1 3 2 4 3 5 4 3 1 4	2 0 1 0

Constraints

In test cases worth 30% of the total points, $T_i \neq 2$.
In all test cases, $1 \leq N \leq 100000$, $1 \leq T_i \leq 4$, $0 < A_i < 1000$.

Power Socket

Time Limit: 1 second

Problem

You have kept playing Tower and Dragons for three hours so the hkoiPhone you borrowed from Dr. Jones is running out of battery. The cafe you are in thinks that the customers (like you!) are spending too much time playing mobile games instead of ordering food and drinks. As a countermeasure, the owner wants to stop customers from charging their phones. Unfortunately, it is hard to prevent this if the power sockets are obvious to the customers. Therefore, he designed a very special wall. The wall has many holes such that a power plug can be inserted in many ways, but only one of them (which is known to the shop owner only) is genuine. Those who do not know the exact location of the genuine socket will have to spend a lot of time trying.

As a programmer, you have already written an app to take a picture of the wall and obtained a 2D representation of the wall. The wall's size is W(width) $\times$ H(height) and there are N 1×1 holes. *Figure 1* and *2* show some examples. The shape of the power plug in upright orientation is given in *Figure 3*.

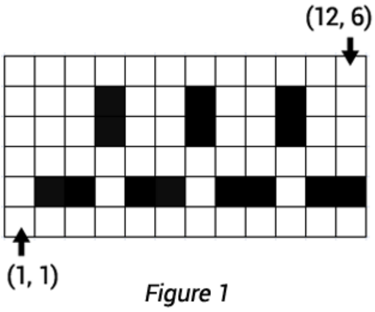


Figure 1

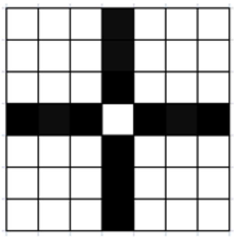


Figure 2

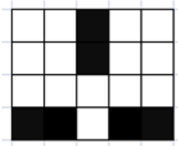


Figure 3

Now, you want to see whether it is worthwhile to try all the possible ways. Write a program to find out how many ways there are to insert the power plug. Please note that the plug can be rotated for insertion.

Input

The first line consists of 3 integers, W, H and N.
The next N lines each contains two integers that represent the coordinates of each hole (X_i, Y_i). The bottom-left square is $(1, 1)$ and the top-right square is (W, H).
The coordinates are given in ascending X followed by Y.

Output

Output one single integer, the number of possible ways to insert the power plug.

Sample test

Input	Output
12 6 14	3
2 2	
3 2	
4 4	
4 5	
5 2	
6 2	
7 4	
7 5	
8 2	
9 2	
10 4	
10 5	
11 2	
12 2	

Input	Output
6 8 12	2
2 1	
2 2	
2 3	
2 4	
2 5	
3 3	
4 3	
5 1	
5 2	
5 3	
5 4	
5 5	

Constraints

In test cases worth 30% of the total points, $5 \leq W, H \leq 1000$ and $6 \leq N \leq 1000$, and the wall is designed such that the plug can only be inserted in the upright orientation.
In test cases worth 50% of the total points, $5 \leq W, H \leq 1000$ and $6 \leq N \leq 1000$.
In test cases worth 70% of the total points, $5 \leq W, H \leq 40000$ and $6 \leq N \leq 2000$.
In test cases worth 100% of the total points, $5 \leq W, H \leq 40000$ and $6 \leq N \leq 50000$.

Fair Santa Claus

Time Limit: 1 second

Problem

It is Christmas time again! Similar to last year, Santa Claus got N presents, which the present labelled i contains i chocolate bars ($i = 1, 2, \dots, N$).

Santa is going to divide and distribute the presents to Alice and Bob. Alice wants to get A presents, while Bob wants to get B presents. Since $A + B = N$, Santa should therefore give every present to either Alice or Bob.

In addition, he should be fair - the actual total number of chocolate bars received by Alice, denoted by a, and the actual total number of chocolate bars received by Bob, denoted by b, should be as close as possible. Santa's goal is to minimize the absolute difference of a and b.

You are going to help Santa divide the presents. If there are more than one optimal solutions, output any one.

Input

The input contains 3 integers in a single line, N, A and B.

Output

The first line should contain A integer(s), the label(s) of the present(s) Alice gets.
The second line should contain B integer(s), the label(s) of the present(s) Bob gets.
The labels may be outputted in any order.

Sample test

Input	Output
5 2 3	3 5 4 1 2

Input	Output
6 4 2	1 2 3 4 6 5

Constraints

In test cases worth 25% of the total points, $2 \le N \le 15$.
In test cases worth 45% of the total points, $2 \le N \le 200$.
In test cases worth 60% of the total points, $2 \le N \le 2000$.
In all test cases, $2 \le N \le 200000$. $1 \le A, B \le N - 1$ and $A + B = N$.

Hint

64-bit integer types may be needed.