Hong Kong Olympiad in Informatics 2017/18 Junior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
J181	Wings and Nuggets	1.000 s	256 MB	22 + 24 + 26 + 28
J182	Rope	1.000 s	256 MB	8 + 18 + 21 + 53
J183	Shortest Path	1.000 s	256 MB	16 + 11 + 25 + 28 + 20
J184	Mysterious Area	1.000 s	256 MB	10 + 17 + 24 + 19 + 30

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.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

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

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long` and its token for `scanf` /`printf` is `%lld`.

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

J181 - WINGS AND NUGGETS

Time Limit: 1.000 s / Memory Limit: 256 MB

One day, Alice and Bob visit the M fast food chain because they love the food there. Alice likes Wings and Bob likes Nuggets. In the restaurant, Wings and Nuggets come in different quantities and they have different prices. Can you help Alice and Bob to find the optimum way to order?

The M restaurant offers Wings in packs of 2 or 4. The price of 2 Wings is W_2 dollars and the price of 4 Wings is W_4 dollars. On the other hand, the restaurant offers Nuggets in packs of 4, 6 or 9. The prices of 4, 6 and 9 Nuggets are N_4, N_6, and N_9 dollars respectively. Note that all prices are positive integers and the price of a larger pack is always greater than that of a smaller pack.

The task is divided into 4 subtasks. You may write separate programs to solve them separately as the scores of the subtasks are cumulative.

SUBTASK 1

In subtask 1, you should help Alice to find the minimum cost of buying at least X Wings.

Input: The first line contains a single integer 1, indicating subtask 1. The second line contains two integers W_2 and W_4, the prices of 2 Wings and 4 Wings respectively. The third line contains an integer X.

Output a single integer: the minimum cost (in dollars) to buy at least X Wings.

SUBTASK 2

In subtask 2, you should help Alice to find the maximum number of Wings she can buy with not more than Y dollars.

Input: The first line contains a single integer 2, indicating subtask 2. The second line contains two integers W_2 and W_4, the prices of 2 Wings and 4 Wings respectively. The third line contains an integer Y.

Output a single integer: the maximum number of Wings she can buy.

SUBTASK 3

In subtask 3, you should help Bob to find the minimum cost of buying at least X Nuggets.

Input: The first line contains a single integer 3, indicating subtask 3. The second line contains three integers N_4, N_6 and N_9, the prices of 4, 6 and 9 Nuggets respectively. The third line contains an integer X.

Output a single integer: the minimum cost (in dollars) to buy at least X Nuggets.

SUBTASK 4

In subtask 4, you should help Bob to find the maximum number of Nuggets he can buy with not more than Y dollars.

Input: The first line contains a single integer 4, indicating subtask 4. The second line contains three integers N_4, N_6 and N_9, the prices of 4, 6 and 9 Nuggets respectively. The third line contains an integer Y.

Output a single integer: the maximum number of Nuggets he can buy.

SAMPLE TESTS

	Input	Output

1	1
	10 18
	5
		28

Alice wants to buy at least 5 Wings. The optimum solution is to buy a pack of 2 Wings and a pack of 4 Wings. The total cost is 10 + 18 = 28 dollars.

2	2
	10 18
	101
		22

Alice wants to buy as many Wings as possible with 101 dollars. The optimum solution is to buy 1 pack of 2 Wings and 5 packs of 4 Wings. The total cost is 100 dollars for 22 Wings.

3	3
	14 21 27
	12
		41

Bob wants to buy at least 12 Nuggets. The optimum solution is to buy ta pack of 4 Nuggets and a pack of 9 Nuggets. The total cost is 14 + 27 = 41 dollars.

4	4
	14 21 27
	46
		13

Bob wants to buy as many Nuggets as possible with 46 dollars. The optimum solution is to buy 1 pack of 4 Nuggets and 1 pack of 9 Nuggets. The total cost is 41 dollars for 13 Nuggets.

SUBTASKS

	Points	Constraints
1	22	$1 \le W_2 < W_4 \le 100$ $1 \le X \le 100$
2	24	$1 \le W_2 < W_4 \le 100$ $1 \le Y \le 10000$
3	26	$1 \le N_4 < N_6 < N_9 \le 100$ $1 \le X \le 100$
4	28	$1 \le N_4 < N_6 < N_9 \le 100$ $1 \le Y \le 10000$

J182 - ROPE

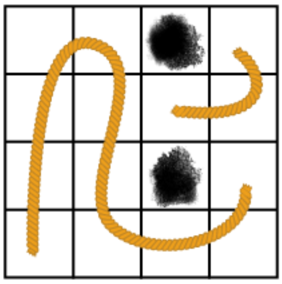
Time Limit: 1.000 s / Memory Limit: 256 MB

After the art lesson, lots of art materials remain in the classroom. Alice picked a huge blank paper and a lengthy rope to play a game.

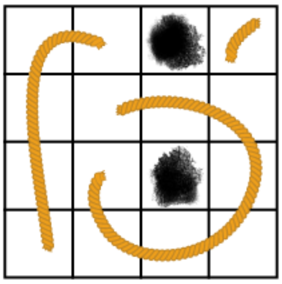
Alice drew a rectangular grid with R rows and C columns on the paper. The rows and columns are numbered from 1 to R and 1 to C respectively. She then painted N of the cells black. As no more ink is left for her to paint the remaining cells, they remain uncolored. However, Alice wants to cover the whole grid with something, so she decides to use the rope to do so.

She can cut the lengthy rope into as many segments as she wants (the segments can be of different lengths), so that she can use the rope segments to cover **ONLY** the uncolored cells. Formally, for each rope segment used, its placement can be expressed as a sequence of **distinct uncolored cells**, where each consecutive pairs of cells in the sequence **must be next to each other** (share an edge). Moreover, the rope segments **MUST NOT** intersect with themselves or each other.

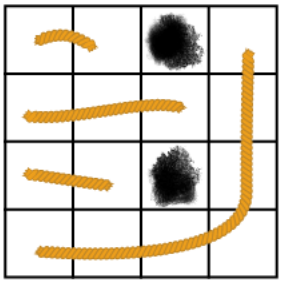
Here are some examples of legal and illegal placements:



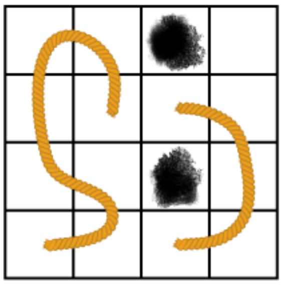
OK
2 ropes used



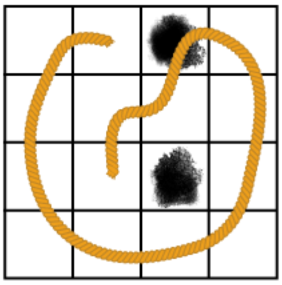
OK
3 ropes used



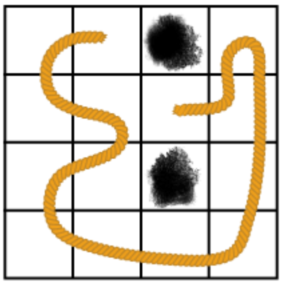
OK
4 ropes used



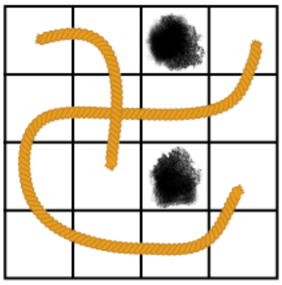
ILLEGAL
Cell not covered



ILLEGAL
Rope covers painted cell



ILLEGAL
Rope overlaps
with itself



ILLEGAL
Rope overlaps
with another rope

To challenge herself, Alice decides to limit the number of rope segments used, so that each uncolored cell should only be covered exactly once. Please help Alice to find any possible way to achieve so.

Here, we denote the number of rope segments you used to be M. Alice will think you have an *excellent* arrangement if $M \leq N + 1$, or a *nice* arrangement if $N + 1 < M \leq R + C + N$.

INPUT

The first line consists of three integers, R, C and N.
Each of the next N lines consists of two integers x and y, stating that one of the painted cells is located on row x, column y. It is guaranteed that $1 \leq x \leq R$, $1 \leq y \leq C$, and the input cells are distinct.

Also, it is guaranteed that an excellent arrangement exists for each test case.

OUTPUT

The first line contains a single integer M, denoting the number of rope segments you use.
For each of the M rope segments, first output a positive integer S, indicating the number of cells covered by this rope segment. In the next S lines, output the sequence of S cells, in separate lines, that this rope segment covers. Each cell should be represented by two integers x and y, stating that the cell is located at row x, column y.

Note: you must follow the order of cells in a sequence when outputting the cells. In other words, each consecutive pairs of cells in the sequence **must be next to each other** (share an edge).

SCORING

For each test case:

- you score zero if the arrangement is illegal, or $M > R + C + N$; otherwise
- you score 60% if the arrangement is *nice*, i.e. $N + 1 < M \leq R + C + N$; otherwise
- you score 100% if the arrangement is *excellent*, i.e. $M \leq N + 1$.

You score for each subtask is the lowest score among all test cases within that subtask.

SAMPLE TESTS

	Input	Output

1	3 3 1
	2 2
		1
		8
		1 1
		1 2
		1 3
		2 3
		3 3
		3 2
		3 1
		2 1

This output scores 100% as it is a legal excellent arrangement using only $M = 1$ rope which is no more than $N + 1 = 2$.

2

4 4 2
1 3
3 3
	3
	1
	1 4
	8
	2 2
	2 3
	2 4
	3 4
	4 4
	4 3
	4 2
	3 2
	5
	1 2
	1 1
	2 1
	3 1
	4 1

This output scores 100% as it is a legal excellent arrangement having $M = 3$, $N = 2$ which fulfills $M \leq N + 1$.
You may refer to the image in the statement (the example with "OK, 3 ropes used").

3

4 4 2
1 3
3 3
	4
	2
	1 1
	1 2
	3
	2 1
	2 2
	2 3
	2
	3 1
	3 2
	7
	4 1
	4 2
	4 3
	4 4
	3 4
	2 4
	1 4

This output scores 60% as it is a legal nice arrangement having $M = 4$ and $N = 2$ which fulfills $N + 1 < M \leq R + C + N$.
You may refer to the image in the statement (the example with "OK, 4 ropes used").

4

2 3 2
1 2
2 2
	2
	2
	1 1
	2 1
	2
	1 3
	2 3

This output scores 100% as it is a legal excellent arrangement having $M = 2$, $N = 2$ which fulfills $M \leq N + 1$.

SUBTASKS

For all cases:
$1 \leq R, C \leq 300$
$0 \leq N < R \times C$

	Points	Constraints
1	8	$R = C = 2$ $N = 0$
2	18	$R = 1$
3	21	$N = 0$
4	53	No additional constraints

J183 - SHORTEST PATH

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice is playing a one-player board game on a $N \times M$ grid. The rows of the grid are numbered from 1 to N, from top to bottom. The columns of the grid are numbered from 1 to M, from left to right. The cell located on the r-th row and the c-th column is labelled (r, c).

Alice has exactly one piece, whose initial position is (S_r, S_c). She would like to bring the piece to the target cell (E_r, E_c) in as few moves as possible.

Before the game starts, a special parameter, K, is chosen. In a move, Alice can choose a positive integer X, such that $X = 1$ or X is an integer multiple of K, and move her piece X cells up, down, left, or right. Alice's piece must remain on the board after each move.

Help Alice find a path from (S_r, S_c) to (E_r, E_c) that requires the minimum number of moves.

INPUT

The first line of input consists of three integers: N, M, and K.

The second line of input consists of two integers: S_r and S_c.

The third line of input consists of two integers: E_r and E_c.

$1 \leq S_r, E_r \leq N$, $1 \leq S_c, E_c \leq M$, $(S_r, S_c) \neq (E_r, E_c)$

OUTPUT

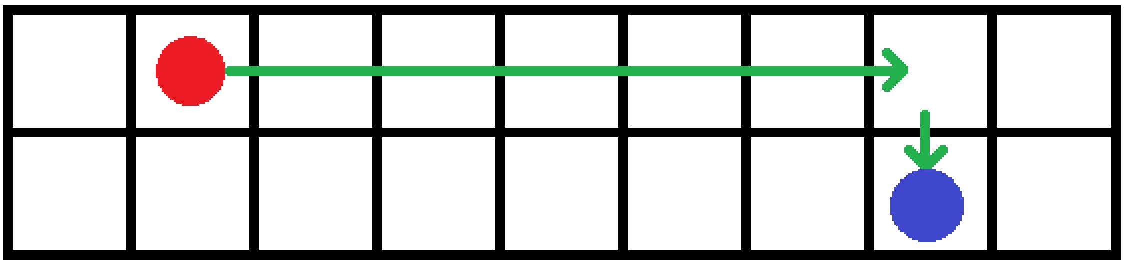
On the first line, output a single integer L, the minimum number of moves required to bring Alice's piece from (S_r, S_c) to (E_r, E_c).

On the next L lines, output an optimal sequence of moves. The i-th line should be of the format `dir X`, where `dir` = `up`, `down`, `left`, or `right`, meaning that, on the i-th move, Alice should move her piece X cells in the direction `dir`.

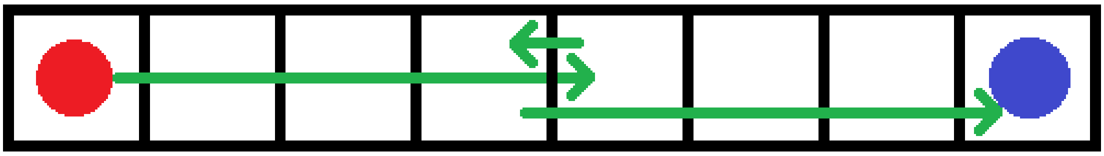
SAMPLE TESTS

	Input	Output

1	2 9 3
	1 2
	2 8
		2
		right 6
		down 1



2	1 8 4
	1 1
	1 8
		3
		right 4
		left 1
		right 4



SUBTASKS

For all cases:
$1 \le N, M \le 10^9$
$1 \le K \le 1000$

	Points	Constraints
1	16	(E_r, E_c) is reachable in one move
2	11	$K = 1$
3	25	$K = 2$
4	28	$(S_r, S_c) = (1, 1)$
5	20	No additional constraints

J184 - MYSTERIOUS AREA

Time Limit: 1.000 s / Memory Limit: 256 MB

In Byteland, there is a bird, it likes to fly around in different places. One day, it accidentally flies into a mysterious area on the sea and it wants to find the way out!

The mysterious area can be represented by a $3N \times N$ grid. There are N rock pillars on the sea. The height of the i^{th} rock pillar is an integer H_i and it occupies cells from $(3i, 1)$ to $(3i, H_i)$, i.e. $(3i, 1)$, $(3i, 2)$, $(3i, 3)$, $\ldots$, $(3i, H_i)$. Because of the mysterious power of the area, all rock pillars have different heights and their heights are between 1 and N (inclusive), which means that $H_{1\ldots N}$ is a permutation of $1 \ldots N$.

The bird is flying through the mysterious area from left (x-coordinate = 1) to right (positive x direction) at height N , i.e. moving at cell $(1, N)$, $(2, N)$, $(3, N)$ and so on. Being affected by the mysterious power of the area, the bird cannot control itself. Whenever the bird bumps into a rock pillar, the mysterious power in the rock will force it to fly down by 1 unit and fly in the opposite direction (i.e. changing the direction of the bird from rightward to leftward or vice versa).

After flying for a long time, the bird will result in two possible situations: it either drops into the sea (moves into a cell with y-coordinate = 0), or successfully escapes from the left side of the mysterious area (moves into a cell with x-coordinate = 0).

You are curious about the trace of the bird. You want to find out whether the bird can successfully escape from the mysterious area or not. If so, state at which height (y-coordinate) the bird flies away. Otherwise, state where (x-coordinate) the bird drops into the sea. Also, find out the total number of cells traversed by the bird before dropping into the sea or escaping from the mysterious area.

INPUT

The first line contains an integer N , the number of rock pillars.
The second line contains N integers, $H_1, H_2, H_3, \ldots, H_N$.

OUTPUT

If the bird can successfully escape from the mysterious area, output `ESCAPE` in the first line and the y-coordinate of the bird when it escapes in the second line.
Otherwise, output `DROP` in the first line and the x-coordinate of where the bird drops into the sea in the second line.
In the third line, output the number of cells traversed by the bird before dropping into the sea or escaping from the mysterious area.

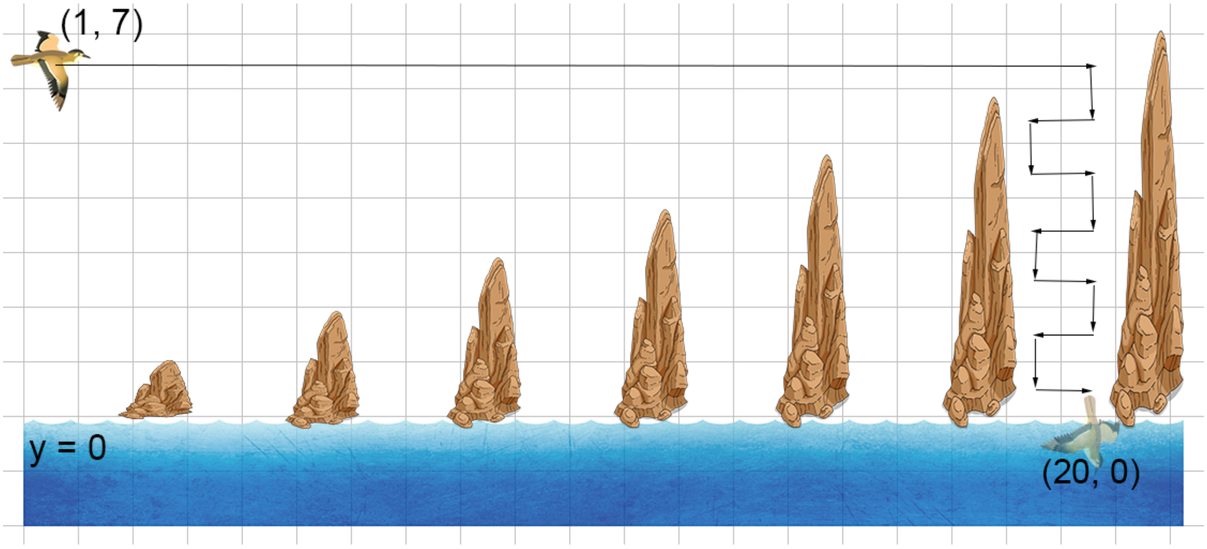
SAMPLE TESTS

Input **Output**

1

```
7
1 2 3 4 5 6 7
```

```
DROP
20
32
```



2

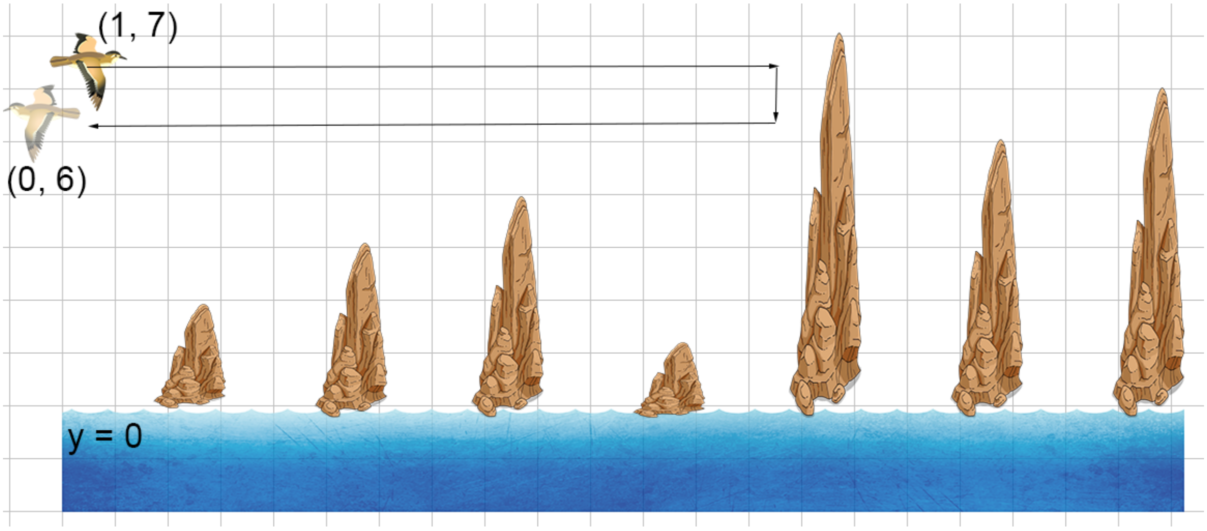
```
7
5 6 4 1 7 2 3
```

```
DROP
11
47
```

3

```
7
2 3 4 1 7 5 6
```

```
ESCAPE
6
28
```



SUBTASKS

For all cases:
$1 \leq N \leq 2 \times 10^5$.
It is guaranteed that $H_{1...N}$ is a permutation of $1 \ldots N$.

	Points	Constraints
1	10	$N = 2$
2	17	$1 \leq N \leq 50$
3	24	$1 \leq N \leq 2000$
4	19	It is guaranteed that the bird can always escape
5	30	No additional constraints