



**CROATIAN NATIONALS 2009**  
**Dubrovnik, March 22 - 26**

**PASCAL/C/C++**  
**Junior category, competition day 2**

---

TASK	KLONOVI	BAGER	PCELICE
input	standard input		
output	standard output		
time limit (per test case)	1 second	2 seconds	2 seconds
memory limit (heap+stack)	64 MB		
points	50	70	80
	200		



Simona breeds sheep. Not being patient enough to wait for sheep to multiply naturally, she built a cloning facility with  $N$  machines in it.

Cloning machines have keyboards that can be used to type in positive integers. If there are  $K$  sheep in a machine and the number  $p$  is typed in, after pressing the button "Clone" cloning starts and soon after  $p \cdot K$  sheep can be found in the machine! For reasons unknown to Simona, cloning only works if the number  $p$  is **prime**. Also, every machine has a maximum capacity – the largest number of sheep that it can hold.

Simona got tired of walking from machine to machine pressing buttons. So she hired  $N$  assistants, one for each machine, and now yells out commands to the assistants:

- **ENTER**  $p$  – every assistant types  $p$  into the keyboard;  **$p$  must be prime**.
- **CLONE**  $a_1 a_2 \dots a_K$  – assistants near machines  $a_1, a_2, \dots, a_K$  all press "Clone". These numbers are **distinct** (so that assistants don't get confused). Because Simona's concentration is limited, she can say **at most  $M$  numbers** in a single CLONE commands.

Simona initially puts **one** sheep into every machine. In order to fully use the machines, Simona wants to fill **every** machine up to its maximum capacity. Once a sequence of commands begins, it is not allowed to take sheep out of machines. Also, Simona wants to yell out **as few commands as possible** to fill all machines with sheep.

Write a program that determines any shortest sequence of commands that will completely fill all machines with sheep.

## **INPUT**

The first line contains the integer  $N$  ( $1 \leq N \leq 50$ ), the number of cloning machines.

The second line contains  $N$  integers, each less than 1 000 000 000. These numbers are the capacities of the machines.

The third line contains the integer  $M$  ( $1 \leq M \leq N$ ), the largest number of machines Simona can trigger in a single CLONE command.

## **OUTPUT**

Output Simona's commands, each on its own line. If multiple solutions exist, output any of them.



## EXAMPLES

### input

```
3
2 3 6
2
```

### output

```
ENTER 2
CLONE 1 3
ENTER 3
CLONE 2 3
```

### input

```
5
25 25 30 25 25
3
```

### output

```
ENTER 2
CLONE 3
ENTER 5
CLONE 1 2 3
CLONE 5 4 1
CLONE 2 4 5
ENTER 3
CLONE 3
```



Inhabitants of neighbouring villages (call the villages A and B) cannot settle their border dispute. The disputed area is rectangular, composed of  $R \times C$  unit cells of land. Every cell has either some number of **apple** trees or some number of **banana** trees growing in it.

A foreign counsellor has been asked to mediate in order to settle the dispute. He has decided that a bulldozer will pass through the area and **destroy** all trees in the cells it goes through. The bulldozer will start in the upper-left corner of the area and always move one cell down, right or diagonally down-right. The bulldozer stops when it reaches the lower-right corner.

The village A will get the land below the bulldozer's path, while village B will get the land above it. Note that it is possible for one of the villages to receive no land at all.

The counsellor noticed that people in village A prefer apples, while people in village B prefer bananas. So he decided to choose the bulldozer's path so that the **sum** of the numbers of apple trees below its path and banana trees above its path is the largest possible.

Write a program that calculates this largest possible sum.

### INPUT

The first line contains the integers R and C ( $2 \leq R, C \leq 1500$ ), the dimensions of the area.

Each of the following R lines contains C descriptions of a cell of land. Each such description consists of the letter 'A' (apples) or 'B' (bananas) and the number of such trees in the cell. Each cell contains between 1 and 99 trees.

### OUTPUT

Output the largest possible sum as described above.

### EXAMPLES

<b>input</b>  4 3 B2 B3 B5 A3 B1 A1 A2 A4 B1 B1 B3 A3  <b>output</b>  17	<b>input</b>  3 5 A5 A2 B3 A6 B2 A1 B20 A5 B3 B6 A3 A5 B3 B8 A3  <b>output</b>  37
--	---

In the first example, the bulldozer should move down-right, down-right and down. There will be  $3+2+4=9$  apple trees below its path and  $3+5=8$  banana trees above the path.



Bees, as we all know, have an important role in pollinating flowers. It is less known that bees have an awkwardly precise protocol for choosing which flowers to pollinate.

There are  $N$  flowers on a meadow. The meadow is a square on the plane with opposite corners in  $(0, 0)$  and  $(M, M)$ , and sides parallel to the coordinate axes. Flowers are points inside the meadow.

A swarm of bees chooses some flower as their base camp and gathers around it. Four warrior bees then take off, one in each of the four cardinal directions: up, down, left and right. Warrior bees stop when they encounter another flower or reach the edge of the meadow.

The rectangle formed by the four warrior bees with sides parallel to the coordinate axes is the area the bees will pollinate. Flowers on the edge of this rectangle will not be pollinated.

Write a program that, given the coordinates of all flowers, **for each flower** calculates the **total number** of pollinated flowers **if that flower** is chosen as the base camp.

### INPUT

The first line contains an integer ( $2 \leq M \leq 1\,000\,000$ ), the dimension of the meadow.

The second line contains an integer  $N$  ( $1 \leq N \leq 300\,000$ ), the number of flowers.

Each of the following  $N$  lines contains two integers  $x$  and  $y$  ( $0 < x, y < M$ ), the coordinates of a flower.

No two flowers will share the same pair of coordinates.

### OUTPUT

For each flower, output the number of pollinated flowers if that flower is chosen as the base camp.

### EXAMPLES

<b>input</b> 3 4 1 2 2 2 2 1 1 1  <b>output</b>  1 1 1 1	<b>input</b> 10 7 1 3 5 3 7 3 4 2 4 4 6 4 6 1  <b>output</b>  3 5 3 5 3 3 5
---	--