# A - Welcome to AtCoder Land

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $100$ points

## Problem Statement

Takahashi is heading to AtCoder Land. There is a signboard in front of him, and he wants to determine whether it says AtCoder Land.

You are given two strings $S$ and $T$ separated by a space. Determine whether $S =$ `AtCoder` and $T =$ `Land`.

## Constraints

- $S$ and $T$ are strings consisting of uppercase and lowercase English letters, with lengths between $1$ and $10$, inclusive.

## Input

The input is given from Standard Input in the following format:

$S$ $T$

## Output

If $S =$ `AtCoder` and $T =$ `Land`, print `Yes`; otherwise, print `No`.

# Sample Input 1

```
AtCoder Land
```

# Sample Output 1

```
Yes
```

$S =$ `AtCoder` and $T =$ `Land`.

---

# Sample Input 2

```
CodeQUEEN Land
```

# Sample Output 2

```
No
```

$S$ is not `AtCoder`.

---

# Sample Input 3

```
aTcodeR lANd
```

## Sample Output 3

```
No
```

Uppercase and lowercase letters are distinguished.

# B - Ticket Counter

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $200$ points

## Problem Statement

At the entrance of AtCoder Land, there is a single ticket booth where visitors line up to purchase tickets one by one. The purchasing process takes $A$ seconds per person. Once the person at the front of the line finishes purchasing their ticket, the next person (if any) immediately starts their purchasing process.

Currently, there is no one in line at the ticket booth, and $N$ people will come to buy tickets one after another. Specifically, the $i$-th person will arrive at the ticket booth $T_i$ seconds from now. If there is already a line, they will join the end of it; if not, they will start the purchasing process immediately. Here, $T_1 < T_2 < \dots < T_N$.

For each $i$ $(1 \leq i \leq N)$, determine how many seconds from now the $i$-th person will finish purchasing their ticket.

## Constraints

- $1 \leq N \leq 100$
- $0 \leq T_1 < T_2 < \cdots < T_N \leq 10^6$
- $1 \leq A \leq 10^6$
- All input values are integers.

---

## Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad A \\
T_1 \quad T_2 \quad \ldots \quad T_N
\end{array}
$$

## Output

Print $N$ lines. The $i$-th line should contain the number of seconds from now that the $i$-th person will finish purchasing their ticket.

---

## Sample Input 1

```
3 4
0 2 10
```

# Sample Output 1

```
4
8
14
```

The events proceed in the following order:

- At $0$ seconds: The 1st person arrives at the ticket booth and starts the purchasing process.
- At $2$ seconds: The 2nd person arrives at the ticket booth and joins the line behind the 1st person.
- At $4$ seconds: The 1st person finishes purchasing their ticket, and the 2nd person starts the purchasing process.
- At $8$ seconds: The 2nd person finishes purchasing their ticket.
- At $10$ seconds: The 3rd person arrives at the ticket booth and starts the purchasing process.
- At $14$ seconds: The 3rd person finishes purchasing their ticket.

---

# Sample Input 2

```
3 3
1 4 7
```

## Sample Output 2

```
4
7
10
```

The events proceed in the following order:

- At $1$ second: The 1st person arrives at the ticket booth and starts the purchasing process.
- At $4$ seconds: The 1st person finishes purchasing their ticket, and the 2nd person arrives at the ticket booth and starts the purchasing process.
- At $7$ seconds: The 2nd person finishes purchasing their ticket, and the 3rd person arrives at the ticket booth and starts the purchasing process.
- At $10$ seconds: The 3rd person finishes purchasing their ticket.

---

## Sample Input 3

```
10 50000
120190 165111 196897 456895 540000 552614 561627 743796 757613 991216
```

## Sample Output 3

```
170190
220190
270190
506895
590000
640000
690000
793796
843796
1041216
```

# C - Popcorn

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $300$ points

## Problem Statement

In AtCoder Land, there are $N$ popcorn stands numbered $1$ to $N$. They have $M$ different flavors of popcorn, labeled $1, 2, \dots, M$, but not every stand sells all flavors of popcorn.

Takahashi has obtained information about which flavors of popcorn are sold at each stand. This information is represented by $N$ strings $S_1, S_2, \dots, S_N$ of length $M$. If the $j$-th character of $S_i$ is o, it means that stand $i$ sells flavor $j$ of popcorn. If it is x, it means that stand $i$ does not sell flavor $j$. Each stand sells at least one flavor of popcorn, and each flavor of popcorn is sold at least at one stand.

Takahashi wants to try all the flavors of popcorn but does not want to move around too much. Determine the minimum number of stands Takahashi needs to visit to buy all the flavors of popcorn.

## Constraints

- $N$ and $M$ are integers.
- $1 \leq N, M \leq 10$
- Each $S_i$ is a string of length $M$ consisting of o and x.
- For every $i$ $(1 \leq i \leq N)$, there is at least one o in $S_i$.
- For every $j$ $(1 \leq j \leq M)$, there is at least one $i$ such that the $j$-th character of $S_i$ is o.

## Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
S_1 \\
S_2 \\
\vdots \\
S_N
\end{array}
$$

## Output

Print the minimum number of stands Takahashi needs to visit to buy all the flavors of popcorn.

---

## Sample Input 1

```
3 5
oooxx
xooox
xxooo
```

## Sample Output 1

```
2
```

By visiting the 1st and 3rd stands, you can buy all the flavors of popcorn. It is impossible to buy all the flavors from a single stand, so the answer is $2$.

---

## Sample Input 2

```
3 2
oo
ox
xo
```

## Sample Output 2

```
1
```

---

## Sample Input 3

```
8 6
xxoxxo
xxoxxx
xoxxxx
xxxoxx
xxoooo
xxxxox
xoxxox
oxoxxo
```

## Sample Output 3

```
3
```

# D - Souvenirs

---

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $350$ points

## Problem Statement

A souvenir shop at AtCoder Land sells $N$ boxes.

The boxes are numbered $1$ to $N$, and box $i$ has a price of $A_i$ yen and contains $A_i$ pieces of candy.

Takahashi wants to buy $M$ out of the $N$ boxes and give one box each to $M$ people named $1, 2, \ldots, M$.

Here, he wants to buy boxes that can satisfy the following condition:

- For each $i = 1, 2, \ldots, M$, person $i$ is given a box containing at least $B_i$ pieces of candy.

Note that it is not allowed to give more than one box to a single person or to give the same box to multiple people.

Determine whether it is possible to buy $M$ boxes that can satisfy the condition, and if it is possible, find the minimum total amount of money Takahashi needs to pay.

## Constraints

- $1 \leq M \leq N \leq 2 \times 10^5$
- $1 \leq A_i, B_i \leq 10^9$
- All input values are integers.

# Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
A_1 \quad A_2 \quad \ldots \quad A_N \\
B_1 \quad B_2 \quad \ldots \quad B_M
\end{array}
$$

# Output

If it is possible to buy $M$ boxes that can satisfy the condition, print the minimum total amount of money Takahashi needs to pay. Otherwise, print $-1$.

---

# Sample Input 1

```
4 2
3 4 5 4
1 4
```

# Sample Output 1

```
7
```

Takahashi can buy boxes $1$ and $4$, and give box $1$ to person $1$ and box $4$ to person $2$ to satisfy the condition.

In this case, he needs to pay $7$ yen in total, and it is impossible to satisfy the condition by paying less than $7$ yen, so print $7$.

---

## Sample Input 2

```
3 3
1 1 1
1000000000 1000000000 1000000000
```

## Sample Output 2

```
-1
```

---

## Sample Input 3

```
7 3
2 6 8 9 5 1 11
3 5 7
```

## Sample Output 3

```
19
```

# E - Alphabet Tiles

---

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $475$ points

## Problem Statement

AtCoder Land sells tiles with English letters written on them. Takahashi is thinking of making a nameplate by arranging these tiles in a row.

Find the number, modulo $998244353$, of strings consisting of uppercase English letters with a length between $1$ and $K$, inclusive, that satisfy the following conditions:

- For every integer $i$ satisfying $1 \leq i \leq 26$, the following holds:
  - Let $a_i$ be the $i$-th uppercase English letter in lexicographical order. For example, $a_1 =$ `A`, $a_5 =$ `E`, $a_{26} =$ `Z`.
  - The number of occurrences of $a_i$ in the string is between $0$ and $C_i$, inclusive.

## Constraints

- $1 \leq K \leq 1000$
- $0 \leq C_i \leq 1000$
- All input values are integers.

---

## Input

The input is given from Standard Input in the following format:

$$K$$
$$C_1 \quad C_2 \quad \ldots \quad C_{26}$$

## Output

Print the answer.

## Sample Input 1

```
2
2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## Sample Output 1

```
10
```

The $10$ strings that satisfy the conditions are `A`, `B`, `C`, `AA`, `AB`, `AC`, `BA`, `BC`, `CA`, `CB`.

## Sample Input 2

```
358
1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## Sample Output 2

```
64
```

## Sample Input 3

```
1000
1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1
000 1000 1000 1000 1000 1000
```

## Sample Output 3

```
270274035
```

# F - Easiest Maze

Time Limit: 2 sec / Memory Limit: 1024 MB

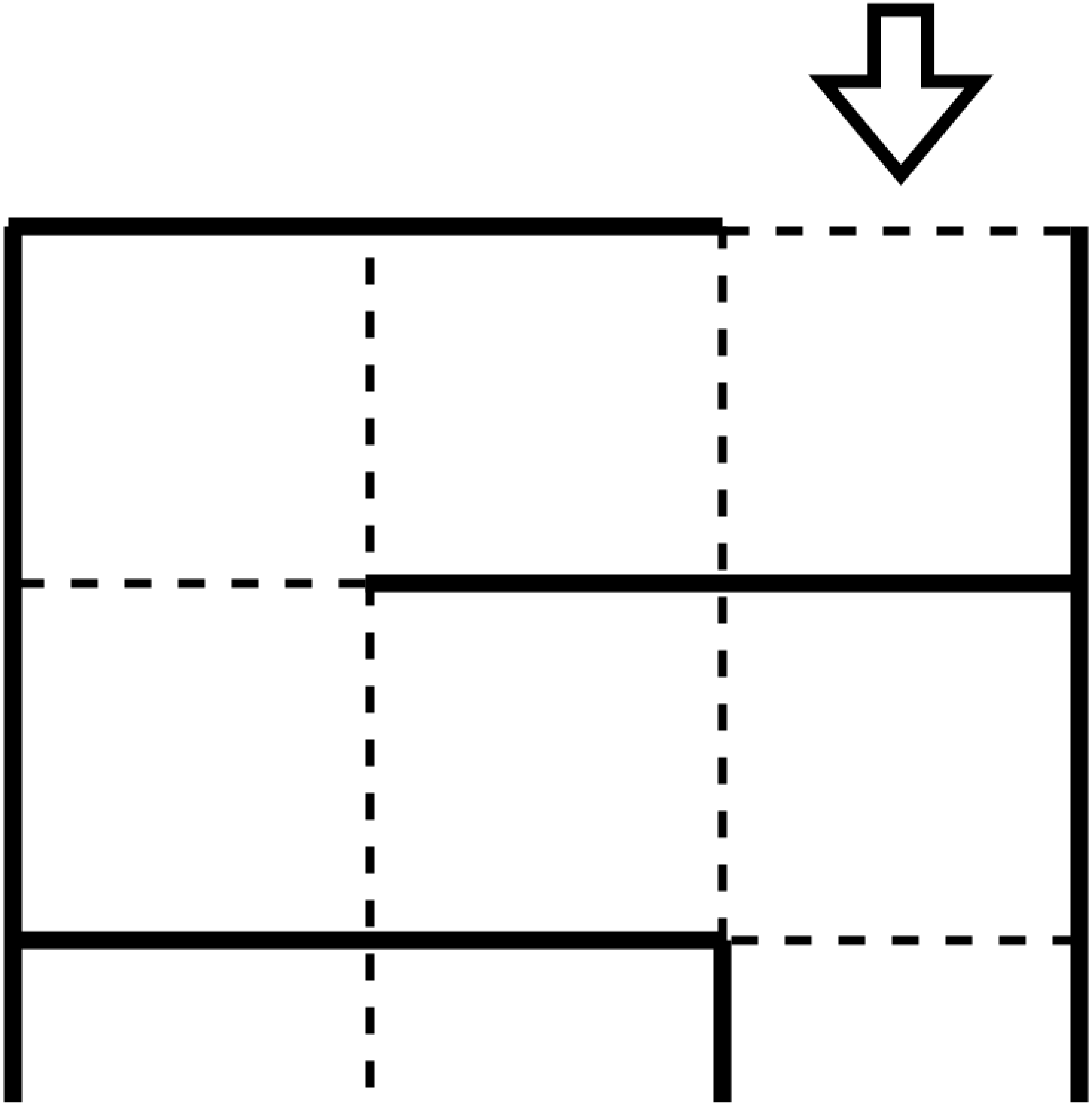
Score : $525$ points

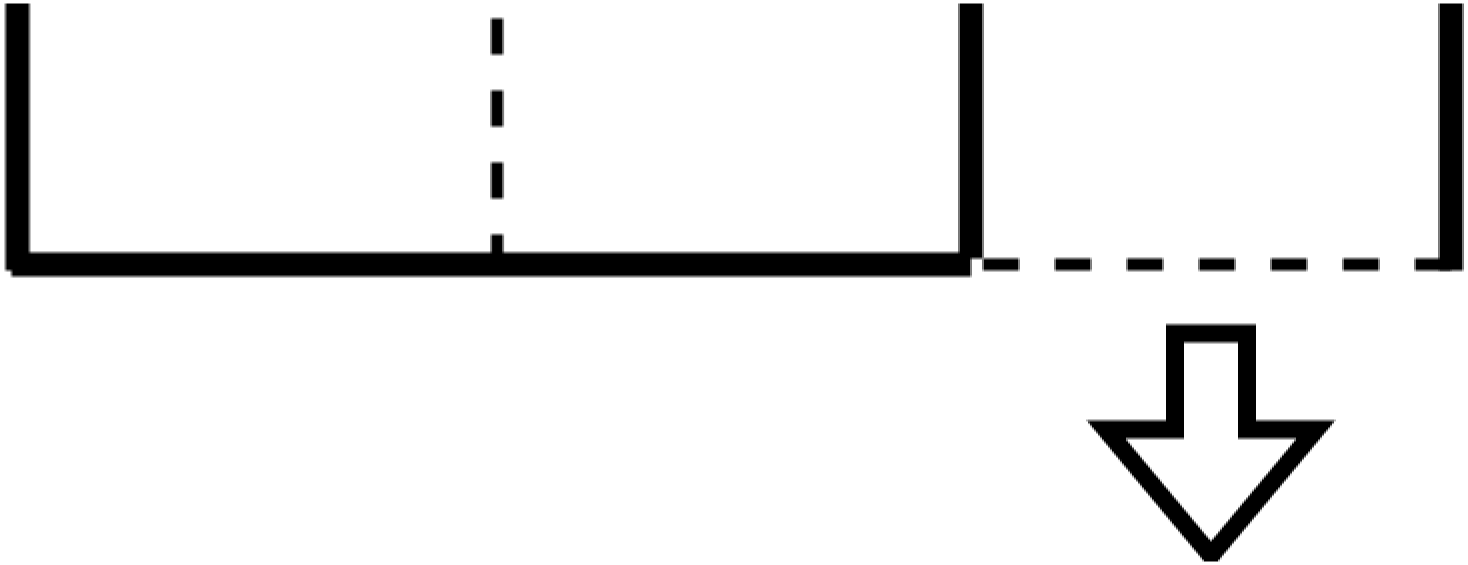
# Problem Statement

Snuke is planning to build a maze as a new attraction in AtCoder Land. The maze is represented as a grid with $N$ rows and $M$ columns, with the top edge of the top-right cell being the entrance and the bottom edge of the bottom-right cell being the exit. He will create the maze by appropriately placing walls between adjacent cells.

He loves simple mazes, so he wants the path from the entrance to the exit to pass through exactly $K$ cells without any branches. Determine if it is possible to create such a maze, and if possible, construct one.

For example, in the following figure, $N = 3$ and $M = 3$, and walls are placed at the solid lines (walls are always placed around the outer perimeter except for the entrance and exit). In this case, the path from the entrance to the exit passes through exactly $7$ cells without any branches.





Below is a formal statement.

There is a grid with $N$ rows and $M$ columns. Let $(i, j)$ denote the cell at the $i$-th row from the top and $j$-th column from the left. For each pair of side-adjacent cells, you can decide whether to place a wall between them. Determine whether it is possible to place walls to satisfy the following condition, and if it is possible, construct one such placement.

> Consider an undirected graph $G$ with $NM$ vertices. Each vertex of $G$ is uniquely labeled by a pair of integers $(i, j)$ $(1 \leq i \leq N, 1 \leq j \leq M)$. Two distinct vertices $(i_1, j_1)$ and $(i_2, j_2)$ are connected by an edge if and only if $|i_1 - i_2| + |j_1 - j_2| = 1$ and there is no wall between the corresponding cells $(i_1, j_1)$ and $(i_2, j_2)$ on the grid.
>
> Condition: there exists a simple path with $K$ vertices that connects the two vertices $(1, M)$ and $(N, M)$, and the connected component containing the vertices $(1, M)$ and $(N, M)$ consists only of this path.

## Constraints

- $2 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq K \leq NM$
- All input values are integers.

---

## Input

The input is given from Standard Input in the following format:

$$N \quad M \quad K$$

# Output

If there is no placement of walls that satisfies the condition, print `No`. Otherwise, print one such placement in the following format. If multiple valid placements exist, any of them can be printed.

**See also the sample outputs below to better understand the complicated output format.**

```
Yes
+++++  ...  +++S+
+o?o?  ...  ?o?o+
+?+?+  ...  +?+?+
+o?o?  ...  ?o?o+
+?+?+  ...  +?+?+
⋮
+o?o?  ...  ?o?o+
+?+?+  ...  +?+?+
+o?o?  ...  ?o?o+
+++++  ...  +++G+
```

Here, `S`, `G`, `+`, and `o` represent the entrance, exit, a wall, and a cell, respectively, and `?` between cells represents a position where a wall can be placed. Replace `?` between two horizontally adjacent cells with `|` if a wall is placed, and `.` otherwise. Replace `?` between two vertically adjacent cells with `-` if a wall is placed, and `.` otherwise.

Below is a formal instruction.

- The output should consist of $2N+2$ lines. Line $1$ should contain the string `Yes`, and lines $2$ to $2N+2$ should contain strings of length $2M+1$ as described below.
  - Line $2$ should be a concatenation of `+` repeated $2M-1$ times, `S`, and `+`, in this order.
  - Line $1+2i$ $(1 \le i \le N)$ should be a concatenation of `+`, `o`, $c_{i,1}$, `o`, $c_{i,2}$, . . ., $c_{i,M-1}$, `o`, `+`, in this order. Here, $c_{i,j}$ is `|` if a wall is placed between cells $(i,j)$ and $(i,j+1)$, and `.` otherwise.

- Line $2+2i\,(1 \leq i \leq N-1)$ should be a concatenation of +, $r_{i,1}$, +, $r_{i,2}$, +, $\ldots$, +, $r_{i,M}$, +, in this order. Here, $r_{i,j}$ is - if a wall is placed between cells $(i,j)$ and $(i+1,j)$, and . otherwise.
- Line $2N+2$ should be a concatenation of + repeated $2M-1$ times, G, and +, in this order.

## Sample Input 1

```
3 3 7
```

## Sample Output 1

```
Yes
+++++S+
+o.o.o+
+.+-+-+
+o.o.o+
+-+-+.+
+o.o|o+
+++++G+
```

This is the same placement of walls as in the figure in the problem statement.

## Sample Input 2

```
3 3 2
```

## Sample Output 2

```
No
```

## Sample Input 3

```
4 1 4
```

## Sample Output 3

```
Yes
+S+
+o+
+.+
+o+
+.+
+o+
+.+
+o+
+G+
```

# G - AtCoder Tour

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : $550$ points

# Problem Statement

AtCoder Land is represented by a grid with $H$ rows and $W$ columns. Let $(i, j)$ denote the cell at the $i$-th row and $j$-th column.

Takahashi starts at cell $(S_i, S_j)$ and repeats the following action $K$ times:

- He either stays in the current cell or moves to an adjacent cell. After this action, if he is in cell $(i, j)$, he gains a fun value of $A_{i,j}$.

Find the maximum total fun value he can gain.

Here, a cell $(x', y')$ is considered adjacent to cell $(x, y)$ if and only if $|x - x'| + |y - y'| = 1$.

# Constraints

- $1 \leq H, W \leq 50$
- $1 \leq K \leq 10^9$
- $1 \leq S_i \leq H$
- $1 \leq S_j \leq W$
- $1 \leq A_{i,j} \leq 10^9$
- All input values are integers.

# Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \quad K \\
S_i \quad S_j \\
A_{1,1} \quad A_{1,2} \quad \ldots \quad A_{1,W} \\
A_{2,1} \quad A_{2,2} \quad \ldots \quad A_{2,W} \\
\vdots \\
A_{H,1} \quad A_{H,2} \quad \ldots \quad A_{H,W}
\end{array}
$$

# Output

Print the answer.

# Sample Input 1

```
2 3 3
1 2
2 1 2
3 4 5
```

## Sample Output 1

```
14
```

Takahashi can gain a total fun value of $14$ by acting as follows:

- Initially, he is at $(1, 2)$.
- He moves to cell $(2, 2)$. Then, he gains a fun value of $A_{2,2} = 4$.
- He moves to cell $(2, 3)$. Then, he gains a fun value of $A_{2,3} = 5$.
- He stays in cell $(2, 3)$. Then, he gains a fun value of $A_{2,3} = 5$.

He cannot gain a total fun value greater than $14$, so print $14$.

---

## Sample Input 2

```
2 2 1000000000
2 1
100 100
100 99
```

## Sample Output 2

```
100000000000
```