

# The Pipeline

□

## The Pipeline

---

You have to lay a pipeline through the great city X. The city is represented by a grid of  $N \times N$  cells. Columns are numbered 1 to  $N$  from left to right. Rows are numbered 1 to  $N$  from top to bottom. In order to lay the pipeline you have to dig in the cells that the pipeline goes through. There is a cost involved with digging up each cell. Your task is to minimize the total cost for digging up cells.

Furthermore pipeline should be laid from left column (column 1) to right column (column  $N$ ) of the city satisfying the following conditions:

1. Since the city is located in the middle of a dessert it doesn't matter from which row the pipeline starts and ends. i.e. The pipeline should start from any cell of column 1 and should end at any cell of column  $N$ .
2. When pipeline enters into the column  $i$  ( $1 \leq i < N$ ), the pipeline can be laid to column  $i+1$  using one of following ways: (lets say pipeline enters the column  $i$  at row  $j$ )
  - Lay the pipeline upward any number of cells (say  $d$ ) within the city boundary and move to column  $i+1$  at row  $j-d$ . i.e.  $d+1$  cells in column  $i$  will be dugged up
  - Lay the pipeline downward any number of cells (say  $d$ ) within the city boundary and move to column  $i+1$  at row  $j+d$ . i.e.  $d+1$  cells in column 1 will be dugged up
  - Lay pipeline to column  $i+1$  through current cell. i.e. Only 1 cell will be dugged up in column  $i$

## Task

---

Given digging up costs for each cell in the city, find the minimum cost for laying the pipeline from column 1 to column  $N$ .

$1 \leq N \leq 1000$

$0 \leq \text{cost of digging up the cell at column } i \text{ and row } j, c(i,j) \leq 1,000,000$

## Input

---

First line of the input will contain an integer  $N$ , the number of rows and column in city.

Next  $N$  lines will represent row 1 to row  $N$  respectively.

Each row will have  $N$  integers separated by spaces representing costs of column 1 to column  $N$  respectively of the corresponding rows.

## Output

---

Output should contain only one integer, minimum cost for digging up the cells that pipeline goes through.  
**Note: There is a newline character at the end of the last line of the output. If you are using C++, use scanf() instead of std::cin as it can be slower reading input**

## Sample Input

---

```
5
1 1 9 1 1
3 1 9 7 1
4 1 9 1 1
5 1 1 1 5
6 1 9 3 1
```

## Sample Output

---

9

## Sample Explanation

---

Pipeline starts from (1,1).  
So pipeline enter the column 1 at (1,1). Moves to column right.  
So pipeline enters the column 2 at (2,1). Moves 4 cells downward and move to column right.  
So pipeline enters the column 3 at (3,4). Moves to column right.  
So pipeline enters the column 4 at (4,4). Move 1 cell upward and move to column right.  
So pipeline enter the column 5 at (5,3)  
**Total cost of the path = 1 + (1 + 1 + 1 + 1) + 1 + (1 + 1) + 1 = 9**

Note: Other paths would produce larger answers. For example, if you consider taking the first row across, then the answer would be 1+1+9+1+1=13, which is larger than the shortest path identified.