

Fall 2020 ECE30017

Problem Solving through Computational Thinking

Week 3

- C2. Ski Trail

Deadline: 11:59 PM, 18 September (Fri)

- P3. Carriers

Deadline: 4:00 PM, 21 September (Mon)

C2. Ski Trail

KOC (Korea Olympic Committee) plans to build a ski resort on the mountains of Pyeongchang. To design ski trails, the KOC engineers modeled the shape of the mountain as a $n \times n$ grid of cells where each cell represents a region of the mountain and the value at the cell represents the altitude of the region.

In this model, a ski trail is represented as a sequence of cells, (s_1, s_2, \dots, s_n) , where the following two conditions hold:

1. s_i and s_{i+1} are adjacent cells in the mountain model.
2. The altitude of s_i is higher than that of s_{i+1} .

To challenge ski players, KOC wants to build a ski trail of which the difficulty is as high as possible. The difficulty of the ski trail is determined as the sum of the altitude differences between two adjacent cells.

Write a program that finds the highest value of difficulty among all possible trails for a given mountain model.

Requirements

- **Input data**

- The first line from the standard input has one integer n , which represents that the size of the model is $n \times n$ for $1 \leq n \leq 500$.
- From the second to the $n + 1$ -th lines, the model of the mountain is given. Each line has n positive integers. The j -th value of the $i + 1$ -th line is the altitude at the (i, j) cell. An altitude is an integer between 0 to 10^8 .

- **Output data**

- Print out one integer, that is, the highest value of ski trail difficulty to the standard output within 0.5 second.

- **Example**

input

```
4
1 6 10 3
5 5 6 1
5 7 2 3
11 5 7 4
```

output

```
9
```

Teams for C2

- Team members must work together for writing a report
 - scores on report and presentation will be shared
 - peer evaluation will be followed
- Each member must construct a solution program individually
 - Team members must not share their program code

| | | | |
|----------|-----|-----|-----|
| Team 201 | 강하영 | 전해주 | |
| Team 202 | 김기훈 | 신희주 | |
| Team 203 | 김석진 | 한찬솔 | |
| Team 204 | 김승우 | 정희석 | |
| Team 205 | 김유진 | 윤지영 | |
| Team 206 | 김윤정 | 이예준 | |
| Team 207 | 김준서 | 정원식 | |
| Team 208 | 김지원 | 임예찬 | |
| Team 209 | 박민준 | 송수근 | |
| Team 210 | 박수현 | 지성민 | |
| Team 211 | 송진범 | 홍원표 | |
| Team 212 | 윤다은 | 최재혁 | |
| Team 213 | 정예은 | 한정섭 | |
| Team 214 | 정진혁 | 홍석현 | |
| Team 215 | 정현섭 | 최우석 | 황소정 |

P3. Carriers

You have N shopping backs from a shopping mall, whose weights are w_1, w_2, \dots, w_N . The weight of a shopping back is no more than M .

To bring these home, you are hiring carriers. A carrier can carry one shopping back by default. And a carrier can carry one more shopping back if the sum of the weights of the two shopping backs is no more than M .

Write a program that finds the minimal number of carriers that can carry all given shopping backs

Requirements

Input

The input data is given to the standard input. The first line has a positive integer M , the maximum weight of a shopping back. After that, an arbitrary number of lines follow, each of which contains a weight of a shopping back.

Output

Print out the minimum number of carriers to the standard output.

Example

Input file

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

Output file

```
3
```