Educational DP Contest AtCoder D - Knapsack 1

Time Limit: 0.25s **Memory Limit:** 64M
Python: 1.0s

These problems are from the AtCoder DP contest, and were transferred onto DMOJ. All problem statements were made by several AtCoder users. As there is no access to the test data, all data is randomly generated. If there are issues with the statement or data, please open a ticket by clicking the "Report an issue" button at the bottom of the page.

There are N items, numbered $1, 2, \ldots, N$. For each i $(1 \leq i \leq N)$, item i has a weight of w_i and a value of v_i.

Taro has decided to choose some of the N items and carry them home in a knapsack. The capacity of the knapsack is W, which means that the sum of the weights of items taken must be at most W.

Find the maximum possible sum of the values of items that Taro takes home.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $1 \leq W \leq 10^5$
- $1 \leq w_i \leq W$
- $1 \leq v_i \leq 10^9$

Input Specification

The first line of input will contain 2 space separated integers, N and W.

The next N lines will contain 2 space separated integers, w_i and v_i, the weight and value of item i.

Output Specification

You are to output a single integer, the maximum possible sum of the values of items that Taro takes home.

Sample Input 1

```
3 8
3 30
4 50
5 60
```

Sample Output 1

```
90
```

Sample Input 2

```
5 5
1 1000000000
1 1000000000
1 1000000000
1 1000000000
1 1000000000
```

Sample Output 2

```
5000000000
```

Sample Input 3

```
6 15
6 5
5 6
6 4
6 6
3 5
7 2
```

Sample Output 3

```
17
```

Sample Explanations

For the first sample, items 1 and 3 should be taken. Then, the sum of the weights is $3 + 5 = 8$, and the sum of the values is $30 + 60 = 90$.

For the second sample, it is important to note that the answer may not fit in a 32-bit integer type.

For the third sample, items 2, 4, and 5 should be taken. Then, the sum of the weights is $5 + 6 + 3 = 14$, and the sum of the values is $6 + 6 + 5 = 17$.

Educational DP Contest AtCoder H - Grid 1

Time Limit: 1.0s **Memory Limit:** 256M

These problems are from the AtCoder DP contest, and were transferred onto DMOJ. All problem statements were made by several AtCoder users. As there is no access to the test data, all data is randomly generated. If there are issues with the statement or data, please open a ticket by clicking the "Report an issue" button at the bottom of the page.

There is a grid with H horizontal rows and W vertical columns. Let (i, j) denote the square at the i-th row from the top and the j-th column from the left.

For each i and j $(1 \leq i \leq H, 1 \leq j \leq W)$. Square (i, j) is described by a character $a_{i,j}$. If $a_{i,j}$is `.`, square (i, j) is an empty square; if $a_{i,j}$ is `#`, square (i, j) is a wall square. It is guaranteed that squares $(1, 1)$ and (H, W) are empty squares.

Taro will start from square $(1, 1)$ and reach (H, W) by repeatedly moving right or down to an adjacent empty square.

Find the number of Taro's paths from square $(1, 1)$ to (H, W). As the answer can be extremely large, find the count modulo $10^9 + 7$.

Constraints

- H and W are integers
- $2 \leq H, W \leq 1000$
- $a_{i,j}$ is `.` or `#`
- Squares $(1, 1)$ and (H, W) are empty squares

Input Specification

The first line will contain 2 space separated integers, H and W.

The next H lines will each contain W characters, either a `.` or `#`.

Output Specification

Print the number of Taro's paths from square $(1, 1)$ to (H, W), modulo $10^9 + 7$.

Sample Input 1

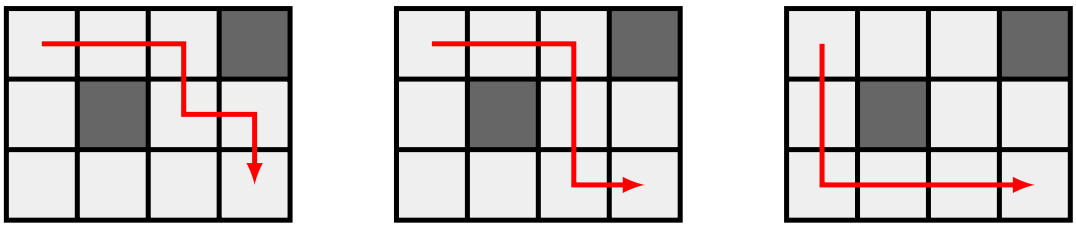
```
3 4
...#
.#..
....
```

Sample Output 1

```
3
```

Explanation For Sample 1

There are three paths as follows:



Sample Input 2

```
5 2
..
#.
..
.#
..
```

Sample Output 2

```
0
```

Explanation For Sample 2

There may be no paths.

Sample Input 3

```
5 5
..#..
.....
#...#
.....
..#..
```

Sample Output 3

```
24
```

Sample Input 4

```
20 20
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
....................
```

Sample Output 4

```
345263555
```

Explanation For Sample 4

Be sure to print the count modulo $10^9 + 7$.

Educational DP Contest AtCoder K - Stones

Time Limit: 1.0s **Memory Limit:** 64M

These problems are from the AtCoder DP contest, and were transferred onto DMOJ. All problem statements were made by several AtCoder users. As there is no access to the test data, all data is randomly generated. If there are issues with the statement or data, please open a ticket by clicking the "Report an issue" button at the bottom of the page.

There is a set $A = \{a_1, a_2, \ldots, a_N\}$ consisting of N positive integers. Taro and Jiro will play the following game against each other.

Initially, we have a pile consisting of K stones. The two players perform the following operation alternately, starting from Taro:

- Choose an element x in A, and remove exactly x stones from the pile.

A player loses when he becomes unable to play. Assuming that both players play optimally, determine the winner.

Constraints

- All values in input are integers.
- $1 \leq N \leq 100$
- $1 \leq K \leq 10^5$
- $1 \leq a_1 < a_2 < \cdots < a_N \leq K$

Input Specification

The first line will contain 2 space separated integers N, K.

The next line will contain N space separated integers, $a_1, a_2, \ldots, a_N$.

Output Specification

If Taro will win, print `First`; if Jiro will win, print `Second`.

Sample Input 1

```
2 4
2 3
```

Sample Output 1

```
First
```

Explanation For Sample 1

If Taro removes three stones, Jiro cannot make a move. Thus, Taro wins.

Sample Input 2

```
2 5
2 3
```

Sample Output 2

```
Second
```

Explanation For Sample 2

Whatever Taro does in his operation, Jiro wins, as follows:

- If Taro removes two stones, Jiro can remove three stones to make Taro unable to make a move.
- If Taro removes three stones, Jiro can remove two stones to make Taro unable to make a move.

Sample Input 3

```
2 7
2 3
```

Sample Output 3

```
First
```

Explanation For Sample 3

Taro should remove two stones. Then, whatever Jiro does in his operation, Taro wins, as follows:

- If Jiro removes two stones, Taro can remove three stones to make Jiro unable to make a move.
- If Jiro removes three stones, Taro can remove two stones to make Jiro unable to make a move.

Sample Input 4

```
3 20
1 2 3
```

Sample Output 4

```
Second
```

Sample Input 5

```
3 21
1 2 3
```

Sample Output 5

```
First
```

Sample Input 6

```
1 100000
1
```

Sample Output 6

```
Second
```

Educational DP Contest AtCoder L - Deque

Time Limit: 1.0s **Memory Limit:** 512M

These problems are from the AtCoder DP contest, and were transferred onto DMOJ. All problem statements were made by several AtCoder users. As there is no access to the test data, all data is randomly generated. If there are issues with the statement or data, please open a ticket by clicking the "Report an issue" button at the bottom of the page.

Taro and Jiro will play the following game against each other.

Initially, they are given a sequence $a = (a_1, a_2, \ldots, a_N)$. Until a becomes empty, the two players perform the following operation alternately, starting from Taro:

- Remove the element at the beginning or the end of a. The player earns x points, where x is the removed element.

Let X and Y be Taro's and Jiro's total score at the end of the game, respectively. Taro tries to maximize $X - Y$, while Jiro tries to minimize $X - Y$.

Assuming that the two players play optimally, find the resulting value of $X - Y$.

Constraints

- All values in input are integers.
- $1 \leq N \leq 3000$
- $1 \leq a_i \leq 10^9$

Input Specification

The first line will contain the integer N.

The next line will contain N integers, $a_1, a_2, \ldots, a_N$.

Output Specification

Print the resulting value of $X - Y$, assuming that the two players play optimally.

Sample Input 1

```
4
10 80 90 30
```

Sample Output 1

```
10
```

Explanation For Sample 1

The game proceeds as follows when the two players play optimally (the element being removed is written bold):

- Taro: (10, 80, 90, **30**) → (10, 80, 90)
- Jiro: (10, 80, **90**) → (10, 80)
- Taro: (10, **80**) → (10)
- Jiro: (**10**) → ()

Here, $X = 30 + 80 = 110$ and $Y = 90 + 10 = 100$.

Sample Input 2

```
3
10 100 10
```

Sample Output 2

```
-80
```

Explanation For Sample 2

The game proceeds, for example, as follows when the two players play optimally:

- Taro: (**10**, 100, 10) → (100, 10)
- Jiro: (**100**, 10) → (10)
- Taro: (**10**) → ()

Here, $X = 10 + 10 = 20$ and $Y = 100$.

Sample Input 3

```
1
10
```

Sample Output 3

```
10
```

Sample Input 4

```
10
1000000000 1 1000000000 1 1000000000 1 1000000000 1 1000000000 1
```

Sample Output 4

```
4999999995
```

Explanation For Sample 4

The answer may not fit into a 32-bit integer type.

Sample Input 5

```
6
4 2 9 7 1 5
```

Sample Output 5

```
2
```

Explanation For Sample 5

The game proceeds, for example, as follows when the two players play optimally:

- Taro: (4, 2, 9, 7, 1, **5**) → (4, 2, 9, 7, 1)
- Jiro: (**4**, 2, 9, 7, 1) → (2, 9, 7, 1)
- Taro: (2, 9, 7, **1**) → (2, 9, 7)
- Jiro: (2, 9, **7**) → (2, 9)
- Taro: (2, **9**) → (2)
- Jiro: (**2**) → ()

Here, $X = 5 + 1 + 9 = 15$, and $Y = 4 + 7 + 2 = 13$.