

A - Two Rectangles

Time Limit: 2 sec / Memory Limit: 256 MiB

Score : 100 points

Problem Statement

There are two rectangles. The lengths of the vertical sides of the first rectangle are A , and the lengths of the horizontal sides of the first rectangle are B . The lengths of the vertical sides of the second rectangle are C , and the lengths of the horizontal sides of the second rectangle are D .

Print the area of the rectangle with the larger area. If the two rectangles have equal areas, print that area.

Constraints

- All input values are integers.
- $1 \leq A \leq 10^4$
- $1 \leq B \leq 10^4$
- $1 \leq C \leq 10^4$
- $1 \leq D \leq 10^4$

Input

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

A B C D

Output

Print the area of the rectangle with the larger area. If the two rectangles have equal areas, print that area.

Sample Input 1

Copy

3 5 2 7

Sample Output 1

Copy

15

The first rectangle has an area of $3 \times 5 = 15$, and the second rectangle has an area of $2 \times 7 = 14$. Thus, the output should be 15, the larger area.

Sample Input 2

[Copy](#)

```
100 600 200 300
```

Sample Output 2

[Copy](#)

```
60000
```

B - Increment Decrement

Time Limit: 2 sec / Memory Limit: 256 MiB

Score : 200 points

Problem Statement

You have an integer variable x . Initially, $x = 0$.

Some person gave you a string S of length N , and using the string you performed the following operation N times. In the i -th operation, you incremented the value of x by 1 if $S_i = \text{I}$, and decremented the value of x by 1 if $S_i = \text{D}$.

Find the maximum value taken by x during the operations (including before the first operation, and after the last operation).

Constraints

- $1 \leq N \leq 100$
- $|S| = N$
- No characters except I and D occur in S .

Input

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

```
 $N$   
 $S$ 
```

Output

Print the maximum value taken by x during the operations.

Sample Input 1

[Copy](#)

```
5
IIDID
```

Sample Output 1

[Copy](#)

```
2
```

After each operation, the value of x becomes 1, 2, 1, 2 and 1, respectively. Thus, the output should be 2, the maximum value.

Sample Input 2

[Copy](#)

```
7
DDIDDII
```

Sample Output 2

[Copy](#)

```
0
```

The initial value $x = 0$ is the maximum value taken by x , thus the output should be 0.

C - Factors of Factorial

Time Limit: 2 sec / Memory Limit: 256 MiB

Score : 300 points

Problem Statement

You are given an integer N . Find the number of the positive divisors of $N!$, modulo $10^9 + 7$.

Constraints

- $1 \leq N \leq 10^3$

Input

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

```
 $N$ 
```

Output

Print the number of the positive divisors of $N!$, modulo $10^9 + 7$.

Sample Input 1

Copy

3

Sample Output 1

Copy

4

There are four divisors of $3! = 6$: 1, 2, 3 and 6. Thus, the output should be 4.

Sample Input 2

Copy

6

Sample Output 2

Copy

30

Sample Input 3

Copy

1000

Sample Output 3

Copy

972926972

D - Walk and Teleport

Time Limit: 2 sec / Memory Limit: 256 MiB

Score : 500 points

Problem Statement

There are N towns on a line running east-west. The towns are numbered 1 through N , in order from west to east. Each point on the line has a one-dimensional coordinate, and a point that is farther east has a greater coordinate value. The coordinate of town i is X_i .

You are now at town 1, and you want to visit all the other towns. You have two ways to travel:

- Walk on the line. Your *fatigue level* increases by A each time you travel a distance of 1, regardless of direction.
- Teleport to any location of your choice. Your fatigue level increases by B , regardless of the distance covered.

Find the minimum possible total increase of your fatigue level when you visit all the towns in these two ways.

Constraints

- All input values are integers.
- $2 \leq N \leq 10^5$
- $1 \leq X_i \leq 10^9$
- For all $i (1 \leq i \leq N - 1)$, $X_i < X_{i+1}$.
- $1 \leq A \leq 10^9$
- $1 \leq B \leq 10^9$

Input

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

```
N  A  B
X1 X2 ... XN
```

Output

Print the minimum possible total increase of your fatigue level when you visit all the towns.

Sample Input 1

[Copy](#)

```
4 2 5
1 2 5 7
```

Sample Output 1

[Copy](#)

```
11
```

From town 1, walk a distance of 1 to town 2, then teleport to town 3, then walk a distance of 2 to town 4. The total increase of your fatigue level in this case is $2 \times 1 + 5 + 2 \times 2 = 11$, which is the minimum possible value.

Sample Input 2

Copy

```
7 1 100
40 43 45 105 108 115 124
```

Sample Output 2

Copy

```
84
```

From town 1, walk all the way to town 7. The total increase of your fatigue level in this case is 84, which is the minimum possible value.

Sample Input 3

Copy

```
7 1 2
24 35 40 68 72 99 103
```

Sample Output 3

Copy

```
12
```

Visit all the towns in any order by teleporting six times. The total increase of your fatigue level in this case is 12, which is the minimum possible value.