NOI 2013 TASKS

Tasks
Task 1: GLOBAL WARMING
Task 2: TRUCKING DIESEL
Task 3: FERRIES

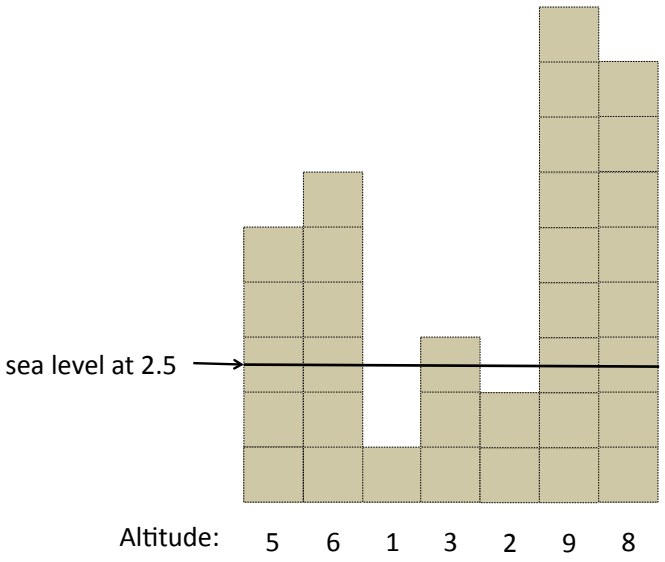
Notes:

1. This document consists of 8 pages including this page.
2. Each task is worth 40 marks.
3. For each task, your program will be tested on a few sets of input instances. We call each set a subtask. Each subtask is worth a few marks. For each subtask, your program either obtains all the marks or none.
4. The subtasks vary in size and complexity, leading to different levels of difficulty. If you find solving the task completely difficult, you may want to focus on the easier subtasks.
5. The maximum execution time on each input instance in Task 1, 2 & 3 is 1.0, 0.5, 1.0 second respectively. Note that there is a limit on the memory size per run. In each run, the *heap* (i.e the memory pool reserved for array, etc) is limited to 32 MB, and the *runtime stack* (i.e. memory storing the routine parameters, etc) is limited to 8 MB.
6. Sample input and output files are provided. For each subtask there is one sample input file and the corresponding output file. The file "`T`***x***`.in.`***y***" is the sample input file for task ***x***, subtask ***y*** (for e.g. "`T1.in.2`" is for task 1, subtask 2) and the file "`T`***x***`.out.`***y***" is the corresponding output file. Filenames ending with extension ".0" (for e.g. "`T1.out.0`") are the corresponding input/output files described in the task statement.
7. Templates for C and Pascal are also provided. You may (but not necessarily) use the templates.

HAPPY PROGRAMMING!

Task 1: GLOBAL WARMING

A scientist wants to study how the rising sea level changes the landscape, in particular, how it changes the number of islands. He first investigates one-dimensional worlds. An one-dimensional world is represented by a sequence of non-negative integers $\langle h_0, h_1, \ldots, h_{n-1} \rangle$, where each integer h_i is the altitude at the location i. The following figure depicts an example of such world represented by the sequence $\langle 5, 6, 1, 3, 2, 9, 8 \rangle$.



Now, if the sea level is at altitude 2.5, there are 3 islands formed by landmass of the first two columns, the fourth column and the last two columns. Furthermore, if the sea level is at altitude 3.5, there are only 2 islands. When the sea level is at altitude x, landmass with altitude x is considered to be submerged under the sea. Hence, if the sea level is at altitude 3, there are 2 islands. Note that having 3 islands is the maximum among all possible sea levels.

Given a one-dimensional world, the scientist wants to find the maximum number of islands among all sea levels.

Input format

Your program must read from the standard input. The first line in the input contains the integer n, the total number of integers in the sequence. Next, it is followed by n lines where each line contains an integer. These n lines represent the sequence $\langle h_0, h_1, \ldots, h_{n-1} \rangle$. All numbers in the sequence are non-negative and smaller than 2^{30}. For the above example, the input is

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

Output format

Your program must write to the standard output an integer, which is the maximum number of islands. For the above example, the output is:

```
3
```

Template

You may use the templates provided. The templates handle the input and output, but without the body of the following subroutines.

- **C program**

```
int gw (int N, int *H);
```

- **Pascal program**

```
function gw (N: LongInt; var H: array of LongInt):  LongInt;
```

Each subroutine takes in two parameters N and H, where N is the size of the array, and H is the array representing the one-dimensional world.

Subtasks

The maximum execution time on each input instance is 1.0 second. Your program will be tested on sets of input instances as follow:

1. (6 marks) All instances in this set satisfy $N \leq 1,000$.
2. (6 marks) All instances in this set satisfy $N \leq 100,000$. In addition, the altitude at each location is at most 20.
3. (7 marks) All instances in this set satisfy $N \leq 100,000$. In addition, the altitude at each location is unique, that is, no two numbers in the input sequence are the same.
4. (10 marks) All instances in this set satisfy $N \leq 1,000,000$. In addition, the numbers in the input sequence are unique.
5. (11 marks) All instances in this set satisfy $N \leq 1,000,000$.

Task 2: TRUCKING DIESEL[1]

Consider a diesel-powered truck that transports diesel fuel from a start position to a destination. The truck does not have a separate fuel tank, but takes its fuel directly from its diesel storage. You need to compute the maximal number of liters of diesel that the truck can bring from the start position to the destination, without using it up along the way. If the truck cannot reach the destination with the given fuel, your answer must be -1.

You are given a map of a terrain in form of an m rows by n columns grid of altitudes. The altitudes are in meters, range from 0 to 4000, and are all divisible by 100. The first row of the grid describes the Northern edge of the terrain, the last row the Southern edge, the first column the Western edge and the last column the Eastern edge. The truck starts at the North-Western corner $(0, 0)$, and its destination is the South-Eastern corner $(m - 1, n - 1)$. In one step, the truck can only go to neighboring positions in the grid, East, South or West. Note that the truck cannot go North and cannot go diagonally.

We assume that diesel fuel has a mass of 1.0 kg per liter. In one step, for each downhill or flat segment, the truck uses 10 liters of diesel. For each uphill segment, the truck needs 4 liters of additional fuel for each 100 m altitude difference of the segment and for each ton[2] of its total mass at the beginning of the segment. In this calculation, we are rounding down the mass of the truck to next lower ton. For example, if the total mass of the truck at the beginning of the segment is 26010 kg, and if the truck moves from an altitude of 400 m to an altitude of 700 m, then the truck will need $10 + 3 \times 4 \times 26 = 322$ liters for that segment. At the end of the segment, the truck will have a mass of 25688 kg.

The truck can dump (i.e throw away) diesel before every step. In the above example, the total mass at the beginning of the segment is 26010 kg. If the truck dumps 11 liters of diesel, its weight will decrease to 25999 kg. Now, the truck will need $10 + 3 \times 4 \times 25 = 310$ liters and at the end of the segment, the truck will have a mass of 25689 kg. Note that the number of liters of diesel that can be dumped must be a whole number. Hence the truck is not allowed to throw away, say 10.1 liter of diesel.

The "unladen" weight of the truck (i.e. weight without fuel) is 8 tons, and it starts out with 25000 liters of diesel fuel.

Example

Let us assume $m = 4$, $n = 4$ and the following altitudes of the grid points.

	0	1	2	3
0	100	200	100	0
1	400	300	100	200
2	200	300	500	500
3	400	400	300	600

[1] This task is a modification of the actual task used in the contest. The original version does not allow dumping of fuel.

[2] Note that one ton is 1000kg.

Note that initially, the truck has a mass of $8000 + 25000 = 33000$ kg. The best course would be to go

- from the start position $(0, 0)$ to $(0, 1)$, requiring 10 liters for the segment, plus $1 \times 4 \times 33$ for the 100 m altitude difference between 100 m and 200 m (fuel consumption: 10 + 132 = 142 liters, mass after the segment: $33000 - 142 = 32858$ kg),
- from $(0, 1)$ to $(1, 1)$, requiring 10 liters for the segment, plus $1 \times 4 \times 32$ liters for the 100 m altitude difference between 200 m and 300 m (fuel consumption: $10 + 128 = 138$ liters, mass after the segment: $32858 - 138 = 32720$ kg),
- from $(1, 1)$ to $(2, 1)$ requiring 10 liters for the segment and no extra fuel (flat segment; fuel consumption: 10 liters, mass after the segment: $32720 - 10 = 32710$ kg),
- from $(2, 1)$ to $(2, 2)$ requiring 10 liters for the segment, plus $2 \times 4 \times 32$ liters for the 200 m altitude difference between 300 m and 500 m (fuel consumption: $10 + 256 = 266$ liters, mass after the segment: $32710 - 266 = 32444$ kg),
- from $(2, 2)$ to $(2, 3)$ requiring 10 liters for the segment and no extra fuel (horizontal segment; fuel consumption: 10 liters, mass after the segment: $32444 - 10 = 32434$ kg),
- from $(2, 3)$ to the destination $(3, 3)$ requiring 10 liters for the segment, plus $1 \times 4 \times 32$ liters for the 100 m altitude difference between 500 m and 600 m (fuel consumption: $10 + 128 = 138$ liters, mass after the segment: $32434 - 138 = 32296$ kg).

The final mass of 32296 kg translates to $32296 - 8000 = 24296$ kg of diesel, which under our assumptions corresponds to 24296 liters of diesel that the truck can transport from the start position to the distination.

Input Format

Your program must read from the standard input. The first line of the input contains m and n. The following m lines contain the altitudes of the grid positions. The example above corresponds to the following file:

```
4 4
100 200 100 0
400 300 100 200
200 300 500 500
400 400 300 600
```

Recall that the altitudes range from 0 to 4000 and are divisible by 100.

Output Format

Your program must write to the standard output an integer, representing the maximal number of liters that can be transported from $(0, 0)$ to $(m-1, n-1)$. If the truck can reach the destination with an empty tank, the number is 0. If the truck cannot reach the destination at all, the number is -1. For the above example, the output is:

```
24296
```

Template

You may use the templates provided. The templates handle the input and output, but without the body of the following subroutines.

- **C program**

```
int truck (int m, int n, int **A);
```

- **Pascal program**

```
function truck (m, n:  LongInt; var A: TwoDarray):  LongInt;
```

The variable `A` is the `m` by `n` two-dimensional array of integers storing the altitudes. The altitude at North-Eastern corner is stored in `A[0][n-1]` and `A[0,n-1]` for the C and Pascal program respectively.

Subtasks

The maximum execution time on each input instance is 0.5 second. Your program will be tested on sets of input instances as follow:

1. (5 marks) $m = 1$, $n \leq 20$, that is, the map is a line and thus the truck can only move East.
2. (5 marks) $m = 2$, $n \leq 20$.
   (Hint: How many times can the truck move South?)
3. (10 marks) $m \leq 20$, $n \leq 20$.
4. (20 marks) $m \leq 1000$, $n \leq 1000$.

Task 3: FERRIES

Kang the Penguin lives on a group of N Antarctic islands, conveniently labelled from 1 to N. Kang's house is on Island 1. He is having a cold today, so he plans to visit a veterinarian who stays on Island N.

Normally, he would swim, but due to his cold, he plans to take ferries to reach his destination instead. There are a total of M ferries (labelled from 1 to M), with Ferry i bringing passengers from some island A_i to some other island B_i for C_i dollars (one direction only). There is at most one ferry going from one island to another island, and some ferries may provide free services. Kang wishes to travel to Island N for as little cost as possible.

Unfortunately for our poor penguin, the captains of the ferries have hatched a money-making scheme today! They know that Kang is planning to take their ferries from Island 1 to Island N, so they conspire to make his journey as expensive as possible. Captains of ferries starting on the same island may permute their destinations among themselves. Due to their contract, however, the cost that a captain charges for riding a ferry remains the same, even if the destination of the ferry has changed. For instance, say that Ferries 1, 2, and 3 start from Island 1. They lead to Islands 2, 3, and 4 respectively, at costs 10, 20, and 30 dollars (also respectively). Then, the captains of Ferry 1 and Ferry 2 may swap destinations, so that now Ferry 1 leads to Island 3 (but still costing 10 dollars) and Ferry 2 leads to Island 2 (but still costing 20 dollars).

The captains, after conspiring, will announce their ferries destinations before Kang boards any ferry, and they cannot change the destinations after the announcement. Kang is aware of the captains' intention but he does not know the ferries destinations before leaving his house. Kang is asking for your help. He wants to know the least amount of money he should bring for the ferries, and yet guaranteed that he has sufficient money to reach his doctor. In other words, he want to find the least possible cost he needs to reach Island N, assuming that the captains make his least cost route as expensive as possible.

Input format

Your program must read from the standard input. The first line of the input contains 2 integers: N and M. The subsequent M lines of the input each contain 3 integers: A_i, B_i, and C_i, representing one ferry each. An example is provided below:

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

Output format

Your program must write to the standard output a single integer, the minimum cost he needs to reach his destination in dollars. For the above example, the output is:

```
9
```

Explanation of Sample Output

Ferries 1, 3 and 5 swap their destinations, so that Ferry 1 now travels to Island 3 (cost still 2 dollars), Ferry 3 to Island 4 (cost still 10 dollars) and Ferry 5 to Island 2 (cost still 7 dollars). After the swaps, the least possible cost for Kang is 9 dollars, traveling either from 1 to 3 to 4, or from 1 to 2 to 4. Note that there is no way to swap the destinations so that the lowest cost from Island 1 to Island 4 is more than 9.

Template

You may use the template provided. The templates handle the input and output, but without the body of the main subroutine.

- **C/C++ program**

```
int ferries(int N, int M, int * A, int * B, int * C)
```

- **Pascal program**

```
function ferries (N, M: LongInt; var A, B, C: array of LongInt):
LongInt;
```

Each subroutine takes in N, M, A, B and C and returns the least possible cost for Kang, where N and M are the number of islands and number of ferries respectively, and A, B, C are arrays representing the origins, the destinations, and the costs of the ferries, respectively.

Subtasks

The maximum execution time on each input instance is 1.0 second. Your program will be tested on 4 sets of input instances as follow:

1. (7 marks) All instances in this set satisfy $2 \leq N \leq 100,000$ and $M = 2N - 4$. There are $N - 2$ ferries leaving from Island 1, going to Island 2, 3, ..., $N - 1$ respectively. There are another $N - 2$ ferries, leaving from Island 2, 3, ..., $N - 1$ respectively, going to Island N.

2. (10 marks) All instances in this set satisfy $2 \leq N \leq 100,000$ and $1 \leq M \leq 300,000$. There are one or more ferries leaving from Island 1. There is exactly one ferry leaving from Islands 2, 3, ... $N - 1$.

3. (11 marks) All instances in this set satisfy $2 \leq N \leq 100,000$ and $1 \leq M \leq 300,000$. In addition, there are no cycles. In other words, once Kang has left a given island by a ferry, there is no possible sequence of ferry rides that will bring him back to that island.

4. (12 marks) All instances in this set satisfy $2 \leq N \leq 100,000$ and $1 \leq M \leq 300,000$.

All instances in all sets satisfy $0 \leq C_i \leq 10,000$ for each C_i.