STEMLoyola Challenges

NO SWEAT ◆ NO GAIN

Union Day Challenge

(Challenge 09)

Begins at **12:30 a.m. EAT** on **Saturday 18th April 2020**

Ends at **11:59 p.m. EAT** on **Sunday 26th April 2020**

The Challenge contains five problems (A, B, C, D, & E) arranged in an increasing order of difficulty. Beginners are expected to solve problems A, B, and C; while experienced students are expected to solve all five problems. Score is determined by both the problems you manage to solve and how sooner you submit a correct solution

Visit contest.stemloyola.org to participate in the challenge and/or view live results

Visit contest.stemloyola.org/register to register, if you do not have an account yet

Visit challenges.stemloyola.org/article/contest-registration for registration instructions

Visit challenges.stemloyola.org to access various resources (guidelines, tutorials, articles, videos, solutions to previous challenges, etc.) that you may need to complete the challenge

Consult Mr. Albert, Mr. Samuel, or other Computer Department teachers with any questions or concerns you may have

Problem A: The Tallest One

Description

You have been given the height of all students in the class. Determine the height of the tallest students.

Input

The first line in the input consists of an integer (3 <= **N** <= 500) specifying the total number of students. The second line contains **N** integers specifying the height of each student in the class in centimeters.

Output

Print the height of the tallest student.

NB: *Kindly note that your solution will be run at least five times. Each time, it will be tested against a different set of input. The first few test cases are given below to help you check your solution. The remaining tests can be seen from the contest page for this problem or the results page after you submit your solution*.

Test 1

Input	Output
6 156 140 181 170 130 170	181

Test 2

Input	Output
10 100 90 120 90 90 90 95 180 160 190	190

Test 3

Input	Output
3 160 90 120	160

Problem B: Behind the Tallest One

Description

All students in a class have randomly arranged themselves in a queue. The class teacher wants to make sure that no student is standing behind the tallest student in class. Help the teacher to count how many students are currently behind the tallest student in the class.

Input

The first line in the input consists of an integer (3 <= **N** <= 500) specifying the total number of students. The second line contains **N** integers specifying the height of each student in the class in centimeters. The first height is for the student who is in front.

Output

Print how many students are behind the tallest student in the class. It is guaranteed that there is only a single tallest student in the class.

NB: *Kindly note that your solution will be run at least five times. Each time, it will be tested against a different set of input. The first few test cases are given below to help you check your solution. The remaining tests can be seen from the contest page for this problem or the results page after you submit your solution*.

Test 1

Input	Output
6 156 140 181 170 130 170	3

Test 2

Input	Output
10 100 90 120 90 90 190 95 180 160 110	4

Test 3

Input	Output
3 160 90 170	0

Problem C: The Best Three

Description

Midterm results are out. The Mathematics teacher, Mrs. Mgulu, wants to select three students who will accompany her to the Parents-Teachers Association discussions. She decided to randomly arrange the Mathematics results in a list and select three students based on the three consecutive students who will have the highest total marks. *Speaking of a teacher who complicates simple tasks!*

Input

The first line in the input consists of an integer (3 <= **N** <= 500) specifying the total number of students. The second line contains **N** integers specifying the score of each student who took the test. The score is between zero and one hundred.

Output

List the scores of the three students who will be chosen by Mrs. Mgulu. It is guaranteed that there is only one way to select the three consecutive students.

NB: *Kindly note that your solution will be run at least five times. Each time, it will be tested against a different set of input. The first few test cases are given below to help you check your solution. The remaining tests can be seen from the contest page for this problem or the results page after you submit your solution*.

Test 1

Input	Output
6 56 40 80 90 30 70	40 80 90

There are only four possibilities: (56, 40, 80), (40, 80, 90), (80, 90, 30), and (90, 30, 70). Of the three, (40, 80, 90) has the highest total marks (210).

Test 2

Input	Output
10 100 90 20 90 90 90 95 80 60 90	90 90 95

Test 3

Input	Output
3 60 90 20	60 90 20

Problem D: The Swaps

Description

All students in a class are standing in a line. The class teacher wants to make sure that each student is standing behind someone is shorter than him/her. The teacher observes the students and each time the teacher sees a student who is standing behind a taller person, the teacher swaps the students. The teacher continues to swap students until all students are standing behind someone who is shorter than themselves. Determine the minimum number of swaps the teacher needs to perform to accomplish the goal.

Input

The first line in the input consists of an integer (3 <= **N** <= 500) specifying the total number of students. The second line contains **N** integers specifying the height of each student in the class in centimeters. The first height is for the student who is in front.

Output

Determine the minimum swaps needed to accomplish the teachers goal.

NB: *Kindly note that your solution will be run at least five times. Each time, it will be tested against a different set of input. The first few test cases are given below to help you check your solution. The remaining tests can be seen from the contest page for this problem or the results page after you submit your solution*.

Test 1

Input	Output
4 156 140 181 170	2

156 student swaps with 140, and 181 students swaps with 170

Test 2

Input	Output
10 100 110 120 140 150 165 175 180 190 192	0

No swaps are needed.

Test 3

Input	Output
3 160 170 90	2

Test 4

Input	Output
10 150 155 160 162 180 163 170 164 165 161	4

Problem E: Mike's Puzzle 2.0

Description

We all know Mike loves puzzles. We have solved his puzzles in the past. It seems like he is getting better in inventing new ones.

In another of his own creation, he designs a square grid and connects the top left corner to the bottom right corner with continuous paths. To make the puzzle easy to solve, the paths do not contain loops, but surely contains more than one path.

In the game, Mike defines the cost of going through the path as the sum of tiles that one passes through. To win the game, one needs to find the path with the lowest cost of moving from the top left corner to the bottom right corner.

1	5	5		5
1		1	2	
8	9		3	2
	7			1
5	2	9	9	1

Figure 1

1	1	1	1	1					1
2				1					1
2				1					1
2				1					1
2				1					1
2				9	9	9			1
2						9	9	9	9
2									9
2									9
2	2	2	2	2	2	2	2	2	2

Figure 2

Input

The first line in the input consists of an integer (2 <= **N** <= 50) specifying the size of the grid. The following **N** lines contain the grid details. Each line corresponds to a single row of the grid. Each row contains **N** numbers (each as 0 or positive number less than 10) corresponding to **N** columns of the grid. 0 denotes no path, and other number denotes a path and its cost. The data in ***Test 1*** correspond to the grid in in ***Figure 1***, while the data in ***Test 2*** correspond to the grid in ***Figure 2***. Blue tiles denote the path that produce the lowest cost.

Output

Your program should compute the lowest cost in the given grid.

NB: *Kindly note that your solution will be run at least five times. Each time, it will be tested against a different set of input. The first few test cases are given below to help you check your solution. The remaining tests can be seen from the contest page for this problem or the results page after you submit your solution*.

Test 1

Input	Output
5 1 5 5 0 5 1 0 1 2 0 8 9 0 3 2 0 7 0 0 1 5 2 9 9 1	21

Test 2

Input	Output
10 1 1 1 1 1 0 0 0 0 1 2 0 0 0 1 0 0 0 0 1 2 0 0 0 1 0 0 0 0 1 2 0 0 0 1 0 0 0 0 1 2 0 0 0 1 0 0 0 0 1 2 0 0 0 9 9 9 0 0 1 2 0 0 0 0 0 9 9 9 9 2 0 0 0 0 0 0 0 0 9 2 0 0 0 0 0 0 0 0 9 2 2 2 2 2 2 2 2 2 2	37