1. Create two classes:

BaseClass
The Rectangle class should have two data fields-width and height of int types. The class should have display()method, to print the width and height of the rectangle separated by space.
DerivedClass
The RectangleArea class is derived from Rectangle class, i.e., it is the sub-class of Rectangle class. The class should have read_input() method, to read the values of width and height of the rectangle. The RectangleArea class should also overload the display() method to print the area (width*height) of the rectangle.
Input Format The first and only line of input contains two space separated integers denoting the width and height
Output Format The output should consist of exactly two lines: In the first line, print the width and height of the rectangle separated by space. In the second line, print the area of the rectangle.
Sample TestCase 1
Input
12 9
Output
12 9
108

2. Monkeys in the garden
In a garden, trees are arranged in a circular fashion with an equal distance between two adjacent trees. The height of trees may vary. Two monkeys live in that garden and they were very close to each other. One day they quarreled due to some misunderstanding. None of them were ready to leave the garden. But each one of them wants that if the other wants to meet him, it should take maximum possible time to reach him, given that they both live in the same garden.

The conditions are that a monkey cannot directly jump from one tree to another. There are 30 trees in the garden. If the height of a tree is H, a monkey can live at any height from 0 to H. Lets say he lives at the height of K then it would take him K unit of time to climb down to the ground level. Similarly, if a monkey wants to climb up to K height it would again take K unit of time. The time to travel between two adjacent trees is 1 unit. A monkey can only travel in a circular fashion in the garden because there is a pond at the center of the garden.

So the question is where should two monkeys live such that the traveling time between them is maximum while choosing the shortest path between them in any direction clockwise or anti-clockwise. You have to answer only the maximum traveling time.

Input Format
The first line consists of total number of trees (N) Each of the following N lines contains the height of trees in a clockwise fashion.

Output Format

You must print an integer which will be the maximum possible travel time.

Sample TestCase 1

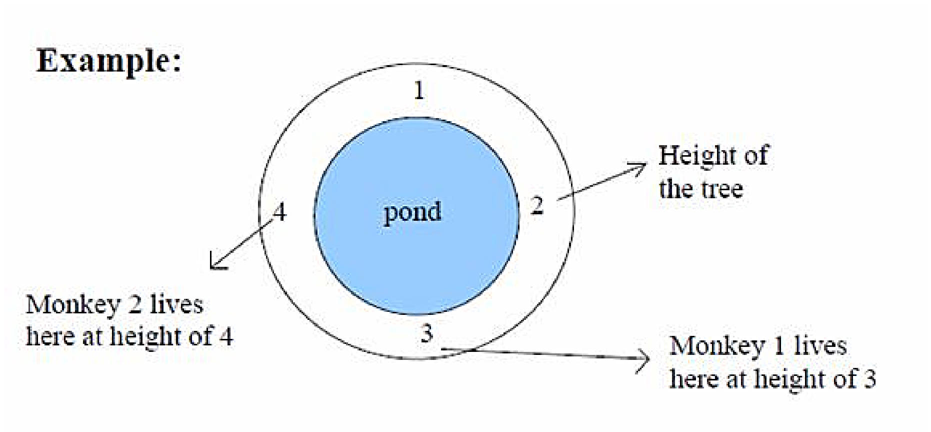
Input

4 1 2 3 4

Output

8

Explanation


Example:
1
2
3
4
pond
Height of the tree
Monkey 2 lives here at height of 4
Monkey 1 lives here at height of 3

In above example if monkey 1 wants to meet monkey 2 it would take total time : 3(climb down the tree) + 1(time between the tree)+ 4(climb up the tree)= **8 unit**.

3. You are given a list of non-negative integers, a1, a2, ..., an, and a target, S. Now you have 2 symbols + and -. For each integer, you should choose one from + and - as its new symbol.

Find out how many ways to assign symbols to make sum of integers equal to target S.

Example 1:

Input: nums is [1, 1, 1, 1, 1], S is 3.
Output: 5
Explanation:

-1+1+1+1+1 = 3
+1-1+1+1+1 = 3
+1+1-1+1+1 = 3
+1+1+1-1+1 = 3
+1+1+1+1-1 = 3

There are 5 ways to assign symbols to make the sum of nums be target 3.

4. Given a matrix of M x N elements (M rows, N columns), return all elements of the matrix in diagonal order as shown in the below image.

Input:
[
 [1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]
]
Output: [1,2,4,7,5,3,6,8,9]

5. Given an n sized unsorted array, find median and mode using counting sort technique. This can be useful when array elements are in limited range.

Input : array a[] = {1, 1, 1, 2, 7, 1}
Output : Mode = 1
Auxiliary(count) array before summing its previous counts, c[]:
Index: 0 1 2 3 4 5 6 7 8 9 10
count: 0 4 1 0 0 0 0 1 0 0 0
Mode = 1

Input : array a[] = {9, 9, 9, 9, 9}
Output : Mode = 9
Auxiliary(count) array before summing its previous counts, c[]:
Index: 0 1 2 3 4 5 6 7 8 9 10
count: 0 0 0 0 0 0 0 0 0 5 0
Mode = 9

6. Given a circular array (the next element of the last element is the first element of the array), print the Next Greater Number for every element. The Next Greater Number of a number x is the first greater number to its traversing-order next in the array, which means you could search circularly to find its next greater number. If it doesn't exist, output -1 for this number
Input: [1,2,1]
Output: [2,-1,2]
Explanation: The first 1's next greater number is 2;
The number 2 can't find next greater number;
The second 1's next greater number needs to search circularly, which is also 2.