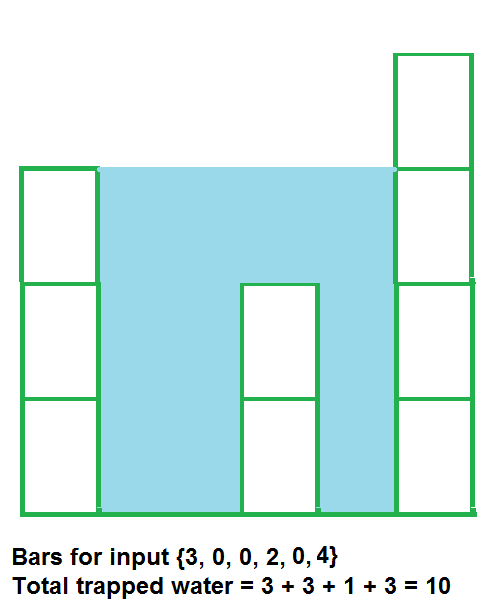
Given an array **arr[]** of **N** non-negative integers representing the height of blocks at index **i** as **Ai** where the width of each block is 1. Compute how much water can be trapped in between blocks after raining.  
**The structure is like below**:  
|  |  
|\_|  
We can trap 2 units of water in the middle gap.



**Input:**  
The first line of input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows. Each test case contains an integer **N** denoting the size of the array, followed by **N**space-separated numbers to be stored in the array.

**Output:**  
Output the total unit of water trapped in between the blocks.

**User Task:**  
The task is to complete the function **trappingWater**() which returns the total amount of water that can be trapped.

**Expected Time Complexity:**O(N).  
**Expected Auxiliary Space:**O(N).

**Constraints:**  
1 <= T <= 100  
3 <= N <= 107  
0 <= Ai <= 108

**Example:  
Input:**  
2  
4  
7 4 0 9  
3  
6 9 9

**Output:**  
10  
0

**Explanation:  
Testcase 1:** Water trapped by the block of height 4 is 3 units, block of height 0 is 7 units. So, the total unit of water trapped is 10 units.

// { Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

int trappingWater(int arr[], int n){

int ans=0;

int \*l=arr;

int \*r=(arr+(n-1));

int left\_max=0;

int right\_max=0;

while(l<r)

{

/\*

\*

\* \*

\* \* \*

\* \* \*

\* \* \* \*

we know that right boundary wall is bigger so lets traverse right as water will trapped somewhere

\*/

if(\*l<\*r)

{

//update left\_max

/\*

#

# #

# #

\*/

if(\*l>=left\_max)

left\_max=\*l;

else

{

/\*

#

# #

# #

\*/

ans+= left\_max - (\*l);

}

l++;

}

else

{

/\*

left wall is bigger

#

#

# #

# # # #

# # # # #

# # # # # #

\*/

if(\*r>=right\_max)

right\_max=\*r;

else{

ans+= right\_max - (\*r);

}

r--;

}

}

return ans;

}

// { Driver Code Starts.

int main(){

int t;

cin >> t;

while(t--){

int n;

cin >> n;

int a[n];

for(int i =0;i<n;i++){

cin >> a[i];

}

cout << trappingWater(a, n) << endl;

}

return 0;

} // } Driver Code Ends