[B - Candies](https://vjudge.net/problem/HackerRank-candies)

 Alice is a kindergarten teacher. She wants to give some candies to the children in her class.  All the children sit in a line and each of them has a rating score according to his or her performance in the class.  Alice wants to give at least 1 candy to each child. If two children sit next to each other, then the one with the higher rating must get more candies. Alice wants to minimize the total number of candies she must buy.

**Example**

She gives the students candy in the following minimal amounts: . She must buy a minimum of *10* candies.

**Function Description**

Complete the *candies* function in the editor below.

candies has the following parameter(s):

* *int n:* the number of children in the class
* *int arr[n]:* the ratings of each student

**Returns**

* *int:* the minimum number of candies Alice must buy

**Input Format**

The first line contains an integer, , the size of .  
Each of the next  lines contains an integer  indicating the rating of the student at position .

**Constraints**

**Sample Input 0**

3

1

2

2

**Sample Output 0**

4

**Explanation 0**

Here 1, 2, 2 is the rating. Note that when two children have equal rating, they are allowed to have different number of candies. Hence optimal distribution will be 1, 2, 1.

**Sample Input 1**

10

2

4

2

6

1

7

8

9

2

1

**Sample Output 1**

19

**Explanation 1**

Optimal distribution will be

**Sample Input 2**

8

2

4

3

5

2

6

4

5

Sample Output 2

12

Explanation 2

Optimal distribution will be 12121212.

### **HackerRank - Candies**

<https://www.hackerrank.com/challenges/candies/problem>

Each child's candies depends on three factors

* Rating of Left, Right and Own
* Candies Given to Left, Right​

Since we have right child and left child values to consider,  It seems similar to solving problem for each index using prefix Sum and suffix Sum.  So thinking in that direction let's try to see if two passes over array can solve the problem.

### **Left to Right Pass (Similar to Prefix Array)**

What if we fill each index with following thought

prefixCandies[i] = Number of candies given to  index i, considering only its left neighbour.

Hence, prefixCandies[i] = 1 + prefixCandies[i-1]   if i has MORE rating than left neighbour.  Otherwise, we can assign only 1 candy to ith child.  So here is what prefixCandies looks like

childrenRatings[] = [4, 6, 4, 5, 6, 2]  
prefixCandies[] = [1, 2, 1, 2, 3, 1]

### **Right to Left Pass (Similar to Suffix Array)**

Now if we apply same concept coming back ward Keeping just right neigbor in mind.

childrenRatings[] = [4, 6, 4, 5, 6, 2]  
suffixCandies[] = [1, 2, 1, 1, 2, 1]

### **Observe and Solve It**

Can we have final solution using data of prefixCandies and suffixCandies arrays?

childrenRatings[] = [4, 6, 4, 5, 6, 2]  
prefixCandies[] = [1, 2, 1, 2, 3, 1]  
suffixCandies[] = [1, 2, 1, 1, 2, 1]  
  
finalSolution[] = [1, 2, 1, 2, 3, 1]