**Maximum Rating Sum**

**QUESTION DESCRIPTION**

Ryan is movie obsessed and has collected a list of movie quality ratings. He wants to watch the largest contiguous list of movies with the highest cumulative ratings possible. To do this, he must calculate the sum of all contiguous subarrays in order to determine the maximum possible subarray sum.

For example, ratings are arr = [-1,3,4,-2,5,-7]. We can see that the highest value contiguous subarray runs from arr[1]-arr[4] and is 3 + 4 + -2 + 5 = 10.

**Function Description**

Complete the function maximumSum in the editor below. It must return a long integer denoting the maximum sum for any contiguous subarray in arr.

maximumSum has the following parameter(s):

arr[arr[0],...arr[n-1]]: an array of integers

**Constraints**

* 1 ≤ n ≤ 10
* −10 ≤ arr[i] ≤ 107

**Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer arr[i].

**Sample Case 0**

**Sample Input 0**

4

-1

-2

1

3

Sample Output 0

4

**Explanation 0**

The maximum sum for any contiguous subarray in [−1, −2, 1, 3] is 1 + 3 = 4.

**Sample Case 1**

**Sample Input 1**

4

1

2

3

4

**Sample Output 1**

10

**Explanation 1**

The maximum sum for any contiguous subarray in [1, 2, 3, 4] is 1 + 2 + 3 + 4 = 10.

**CANDIDATE ANSWER**

**Language used: Java 8**

class Result {

/\*

\* maximumSum function

\*

\* return a LONG\_INTEGER maxSoFar.

\* @param acepts list of integer as a param.

\*/

public static long maximumSum(List<Integer> arr) {

// complete the logic

}