Problem statement

You are given an array *'arr'* of length *'n'*, consisting of integers.

A subarray is a contiguous segment of an array. In other words, a subarray can be formed by removing 0 or more integers from the beginning and 0 or more integers from the end of an array**.**

Find the sum of the subarray **(including empty subarray)** having maximum sum among all subarrays.

The sum of an empty subarray is 0.

Example :

```
Input: 'arr' = [1, 2, 7, -4, 3, 2, -10, 9, 1]

Output: 11

Explanation: The subarray yielding the maximum sum is [1, 2, 7, -4, 3, 2].
```

Detailed explanation (Input/output format, Notes, Images)

Sample Input 1 :

```
9
1 2 7 -4 3 2 -10 9 1
```

Sample Output 1 :

```
11
```

Explanation for Sample 1 :

The subarray yielding the maximum sum is [1, 2, 7, -4, 3, 2].

Sample Input 2 :

```
6
10 20 -30 40 -50 60
```

Sample Output 2 :

```
60
```

Sample Input 3 :

```
3
-3 -5 -6
```

Sample Output 3 :

0

Expected time complexity :

The expected time complexity is O(n).

Constraints :

```
1 <= 'n' <= 10 ^ 6
-10 ^ 6 <= 'arr[i]' <= 10 ^ 6
```

```
Time limit: 1sec
```