7. Given a circular integer array nums of length n, return the maximum possible sum of a non-empty subarray of nums.A circular array means the end of the array connects to the beginning of the array. Formally, the next element of nums[i] is nums[(i + 1) % n] and the previous element of nums[i] is nums[(i - 1 + n) % n].A subarray may only include each element of the fixed buffer nums at most once. Formally, for a subarray nums[i], nums[i + 1], ..., nums[j], there does not exist i <= k1, k2 <= j with k1 % n == k2 % n.

```
def maxSubarraySumCircular(nums):
    def kadane(array):
        current_max = global_max = array[0]
        for num in array[1:]:
            current_max = max(num, current_max + num)
            global_max = max(global_max, current_max)
        return global_max
    max_kadane = kadane(nums
    total_sum = sum(nums)
    nums_inverted = [-num for num in nums]
    max_inverted_kadane = kadane(nums_inverted)
    max_circular = total_sum + max_inverted_kadane
    if max_circular == 0:
        return max_kadane
    return max(max_kadane, max_circular)
nums = [5, -3, 5]
print(maxSubarraySumCircular(nums))
```

INPUT:[5,-3,5]

TIME COMPLEXITY:O(n)

OUTPUT: