27. **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.**

Program:

```
def maxCircularSum(a, n):
    if (n == 1):
        return a[0]
    sum = 0
    for i in range(n):
        sum += a[i]
    curr_max = a[0]
    max_so_far = a[0]
    curr_min = a[0]
    min_so_far = a[0]

    for i in range(1, n):

        curr_max = max(curr_max + a[i], a[i])
        max_so_far = max(max_so_far, curr_max)



        curr_min = min(curr_min + a[i], a[i])
        min_so_far = min(min_so_far, curr_min)
    if (min_so_far == sum):
        return max_so_far
    return max(max_so_far, sum - min_so_far)
a = [11, 10, -20, 5, -3, -5, 8, -13, 10]
n = len(a)
```

print("Maximum circular sum is", maxCircularSum(a, n))

Output: