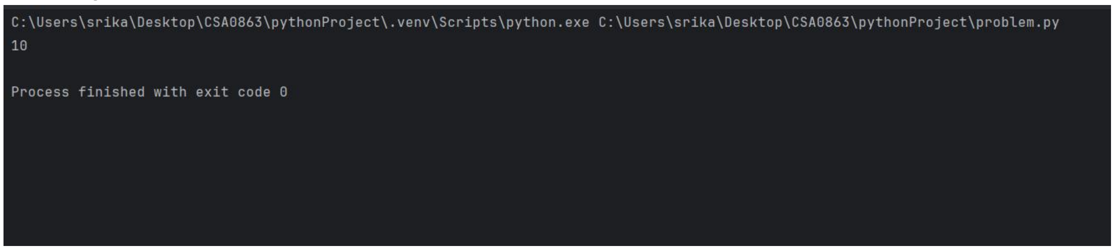
Q).Given a circular integer array nums of length n, return the maximum possible sum of a non-empty subarray of nums.A circular array means the end of the array connects to the beginning of the array. Formally, the next element of nums[i] is nums[(i + 1) % n] and the previous element of nums[i] is nums[(i - 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 maxSubarraySumCircular(nums):
    def kadane(arr):
        max_end_here = max_so_far = arr[0]
        for num in arr[1:]:
            max_end_here = max(num,
max_end_here + num)
            max_so_far = max(max_so_far,
max_end_here)
        return max_so_far
    total_sum = sum(nums)
    max_kadane = kadane(nums)
    inverted_nums = [-num for num in nums]
    max_inverted_kadane =
kadane(inverted_nums)
    min_kadane = -max_inverted_kadane
    if max_kadane < 0:
        return max_kadane
    return max(max_kadane, total_sum -
min_kadane)
nums = [5, -3, 5]
print(maxSubarraySumCircular(nums))
```

Output:



```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe C:\Users\srika\Desktop\CSA0863\pythonProject\problem.py
10

Process finished with exit code 0
```

Time complexity:O(n)