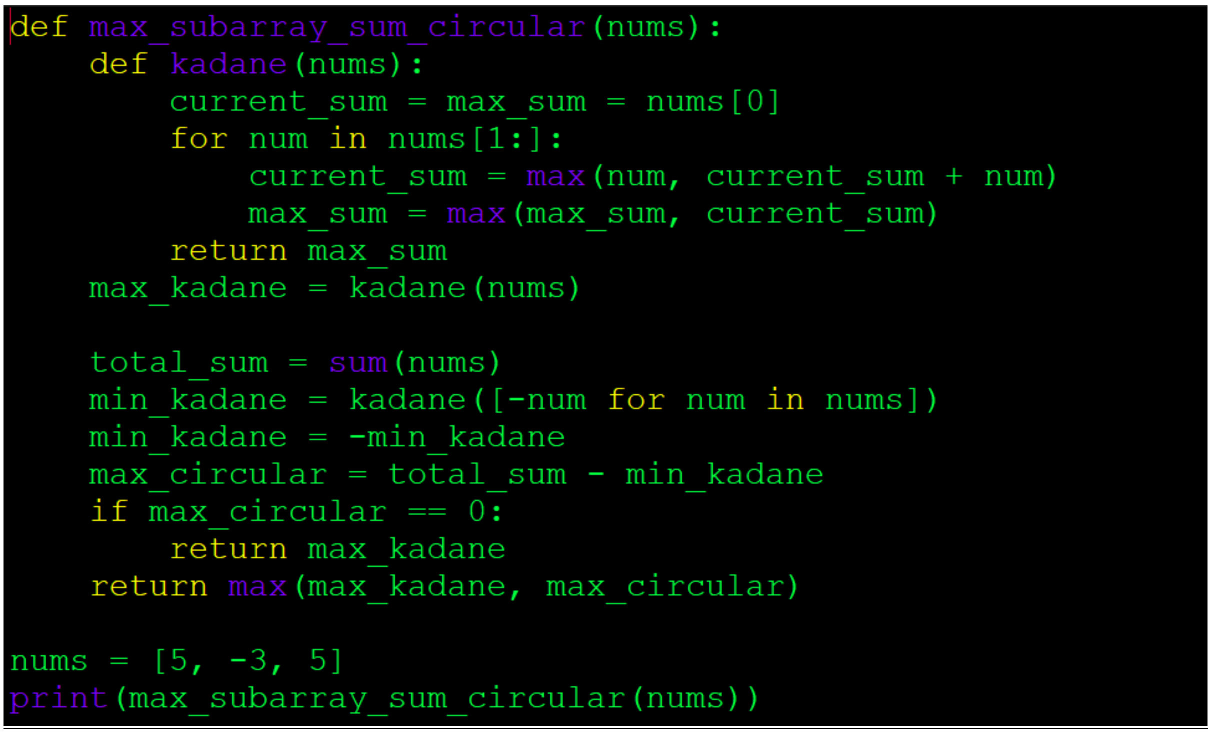
27. Given a circular integer array nums of length n, return the maximum possible sum of a non-empty subarray of nums.A circular array means the end of the array connects to the beginning of the array. Formally, the next element of nums[i] is nums[(i + 1) % n] and the previous element of nums[i] is nums[(i - 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.

Code:

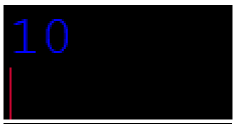


```python
def max_subarray_sum_circular(nums):
    def kadane(nums):
        current_sum = max_sum = nums[0]
        for num in nums[1:]:
            current_sum = max(num, current_sum + num)
            max_sum = max(max_sum, current_sum)
        return max_sum
    max_kadane = kadane(nums)

    total_sum = sum(nums)
    min_kadane = kadane([-num for num in nums])
    min_kadane = -min_kadane
    max_circular = total_sum - min_kadane
    if max_circular == 0:
        return max_kadane
    return max(max_kadane, max_circular)

nums = [5, -3, 5]
print(max_subarray_sum_circular(nums))
```

Output:


```
10
```

Time Complexity:

- T(n)= O(n)