Exercise-56 Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit Given an array of integers nums and an integer limit, return the size of the longest non-empty subarray such that the absolute difference between any two elements of this subarray is less than or equal to limit. Example 1: Input: nums = [8,2,4,7], limit = 4 Output: 2 Explanation: All subarrays are: [8] with maximum absolute diff |8-8| = 0 <= 4. [8,2] with maximum absolute diff |8-2| = 6 > 4. [8,2,4] with maximum absolute diff |8-2| = 6 > 4. [8,2,4,7] with maximum absolute diff |8-2| = 6 > 4. [2] with maximum absolute diff |2-2| = 0 <= 4. [2,4] with maximum absolute diff |2-4| = 2 <= 4. [2,4,7] with maximum absolute diff |2-7| = 5 > 4. [4] with maximum absolute diff |4-4| = 0 <= 4. [4,7] with maximum absolute diff |4-7| = 3 <= 4. [7] with maximum absolute diff |7-7| = 0 <= 4. Therefore, the size of the longest subarray is 2. Example 2: Input: nums = [10,1,2,4,7,2], limit = 5 Output: 4 Explanation: The subarray [2,4,7,2] is the longest since the maximum absolute diff is |2-7| = 5 <= 5. Example 3: Input: nums = [4,2,2,2,4,4,2,2], limit = 0 Output: 3

Program;

```
from collections import deque


def longest_subarray(nums, limit):
    max_queue = deque()
    min_queue = deque()
    left = 0
    result = 0

    for right, num in enumerate(nums):
        while max_queue and num > max_queue[-1]:
            max_queue.pop()
        while min_queue and num < min_queue[-1]:
            min_queue.pop()
        max_queue.append(num)
        min_queue.append(num)

        while max_queue[0] - min_queue[0] > limit:
            if max_queue[0] == nums[left]:
                max_queue.popleft()
            if min_queue[0] == nums[left]:
                min_queue.popleft()
```

```
            left += 1

        result = max(result, right - left + 1)

    return result

# Test cases
print(longest_subarray([8,2,4,7], 4))
print(longest_subarray([10,1,2,4,7,2], 5))
print(longest_subarray([4,2,2,2,4,4,2,2], 0))
```

output;

```
===
2
4
3
```

Time complexity; O(n)