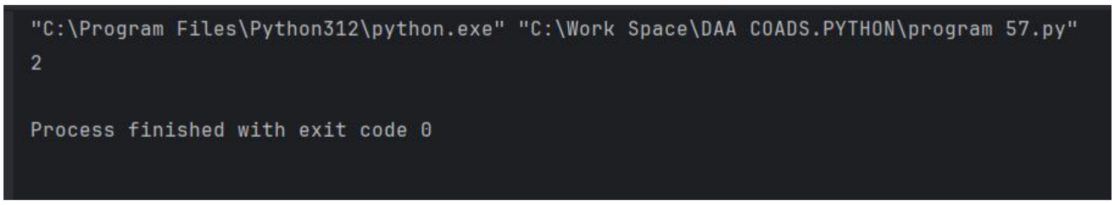
Program 57. Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit Given an array of integers nums and an integer limit, return the size of the longest non-empty subarray such that the absolute difference between any two elements of this subarray is less than or equal to limit. Example 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.

Program:

```
from collections import deque

def longestSubarray(nums, limit):
    max_deque = deque()  # To keep track of the maximum values
    min_deque = deque()  # To keep track of the minimum values
    left = 0  # Left pointer of the sliding window
    max_len = 0  # Resultant maximum length of the subarray

    for right in range(len(nums)):
        # Maintain the decreasing order in max_deque
        while max_deque and nums[max_deque[-1]] <= nums[right]:
            max_deque.pop()
        max_deque.append(right)

        # Maintain the increasing order in min_deque
        while min_deque and nums[min_deque[-1]] >= nums[right]:
            min_deque.pop()
        min_deque.append(right)

        # Check the current window
        while nums[max_deque[0]] - nums[min_deque[0]] > limit:
            left += 1
            if max_deque[0] < left:
                max_deque.popleft()
            if min_deque[0] < left:
                min_deque.popleft()

        # Update the maximum length of the subarray
        max_len = max(max_len, right - left + 1)

    return max_len

# Example usage
nums = [8, 2, 4, 7]
limit = 4
print(longestSubarray(nums, limit))  # Output: 2
```

Output:



```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 57.py"
2

Process finished with exit code 0
```

Time complexity:
O(n)