```c
#include <stdio.h>

int findFirstOccurrence(int nums[], int size, int target) {
    int low = 0;
    int high = size - 1;
    int first_occurrence = -1;

    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (nums[mid] == target) {
            first_occurrence = mid;
            high = mid - 1;
        } else if (nums[mid] < target) {
            low = mid + 1;
        } else {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array: {5, 7, 7, 8, 8, 8, 10, 10, 12}

Target 8 found.
Index of first occurrence: 3
Index of last occurrence: 5
First occurrence value: 8
Last occurrence value: 8
Result: 3, 5
-----------------------------------
Target 7 found.
Index of first occurrence: 1
Index of last occurrence: 2
First occurrence value: 7
Last occurrence value: 7
Result: 1, 2
-----------------------------------
Target 6 not present.
Index of first occurrence: -1
Index of last occurrence: -1
Result: -1, -1
-----------------------------------
Target 12 found.
Index of first occurrence: 8
Index of last occurrence: 8
First occurrence value: 12
Last occurrence value: 12
Result: 8, 8
-----------------------------------
Target 5 found.
Index of first occurrence: 0
Index of last occurrence: 0
First occurrence value: 5
Last occurrence value: 5
Result: 0, 0
```

```c
#include <stdio.h>

int findCeilIndex(int arr[], int size, int x) {
    if (size == 0) {
        return -1;
    }

    int low = 0;
    int high = size - 1;
    int ceil_index = -1;

    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (arr[mid] == x) {
            ceil_index = mid;
            high = mid - 1;
        } else if (arr[mid] > x) {
            ceil_index = mid;
            high = mid - 1;
        } else {
            low = mid + 1;
        }
    }

    return ceil_index;
}

int main() {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array: {2, 5, 5, 8, 12, 12, 15, 20}

Target x = 5
Ceiling Index: 1 (Value: 5)
---------------------------------------
Target x = 9
Ceiling Index: 4 (Value: 12)
---------------------------------------
Target x = 20
Ceiling Index: 7 (Value: 20)
---------------------------------------
Target x = 25
Ceiling Index: -1 (Value: -1)
---------------------------------------
Target x = 1
Ceiling Index: 0 (Value: 2)
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

int findPivotIndex(int arr[], int size) {
    if (size == 0) {
        return -1;
    }

    long long total_sum = 0;
    for (int i = 0; i < size; i++) {
        total_sum += arr[i];
    }

    long long left_sum = 0;

    for (int i = 0; i < size; i++) {
        long long right_sum = total_sum - left_sum - arr[i];

        if (left_sum == right_sum) {
            return i;
        }

        left_sum += arr[i];
    }

    return -1;
}

int main() {
    int arr1[] = {1, 7, 3, 6, 5, 6};
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array 1: {1, 7, 3, 6, 5, 6}
Leftmost Pivot Index: 3
-----------------------------------
Array 2: {1, 2, 3}
Leftmost Pivot Index: -1
-----------------------------------
Array 3: {2, 1, -1}
Leftmost Pivot Index: 0
-----------------------------------
Array 4: {1, 1, 1, 0, 1, 1, 1}
Leftmost Pivot Index: 3
-----------------------------------
Array 5: {0, 0, 0, 0, 10, -10}
Leftmost Pivot Index: 0
-----------------------------------
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

long long sum_to_n(long long n) {
    if (n < 0) return 0;
    return n * (n + 1) / 2;
}

int findPivotInteger(int n) {
    if (n <= 0) {
        return -1;
    }


    long long total_sum = sum_to_n((long long)n);


    long long required_square = total_sum;


    double root_double = sqrt((double)required_square);
    long long x = (long long)root_double;

    if (x * x == required_square && x >= 1 && x <= n) {
        return (int)x;
    }

    return -1;
}

int main() {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
000gn/T/"tempCodeRunnerFile
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Input n = 8, Pivot Integer x = 6
-------------------------------------
Input n = 4, Pivot Integer x = -1
-------------------------------------
Input n = 7, Pivot Integer x = -1
-------------------------------------
Input n = 1, Pivot Integer x = 1
-------------------------------------
Input n = 49, Pivot Integer x = 35
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

int majority_element(int* nums, int n) {
    int candidate = -1;
    int count = 0;

    for (int i = 0; i < n; i++) {
        if (count == 0) {
            candidate = nums[i];
            count = 1;
        } else if (nums[i] == candidate) {
            count++;
        } else {
            count--;
        }
    }

    count = 0;
    for (int i = 0; i < n; i++) {
        if (nums[i] == candidate) {
            count++;
        }
    }

    if (count > n / 2) {
        return candidate;
    } else {
        return -1;
    }
}

int main() {
    int nums1[] = {2, 2, 1, 1, 1, 2, 2};
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array: [2, 2, 1, 1, 1, 2, 2], Majority Element: 2
Array: [1, 2, 3, 4], Majority Element: -1
Array: [3, 3, 2], Majority Element: 3
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

void find_next_greater_element(int arr[], int n, int result[]) {
    int* stack = (int*)malloc(n * sizeof(int));
    int top = -1;

    for (int i = n - 1; i >= 0; i--) {
        while (top != -1 && stack[top] <= arr[i]) {
            top--;
        }

        if (top == -1) {
            result[i] = -1;
        } else {
            result[i] = stack[top];
        }

        top++;
        stack[top] = arr[i];
    }

    free(stack);
}

int main() {
    int arr1[] = {4, 5, 2, 25};
    int n1 = sizeof(arr1) / sizeof(arr1[0]);
    int result1[n1];
    find_next_greater_element(arr1, n1, result1);

    printf("Input Array: [4, 5, 2, 25]\n");
    printf("NGE Array:   ");
    for (int i = 0; i < n1; i++) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
000gn/T/"tempCodeRunnerFile
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Input Array: [4, 5, 2, 25]
NGE Array:   5, 25, 25, -1

Input Array: [13, 7, 6, 12]
NGE Array:   -1, 12, 12, -1
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

void find_previous_greater_element(int arr[], int n, int result[]) {
    int* stack = (int*)malloc(n * sizeof(int));
    int top = -1;

    for (int i = 0; i < n; i++) {
        while (top != -1 && stack[top] <= arr[i]) {
            top--;
        }

        if (top == -1) {
            result[i] = -1;
        } else {
            result[i] = stack[top];
        }

        top++;
        stack[top] = arr[i];
    }

    free(stack);
}

int main() {
    int arr1[] = {15, 10, 18, 12, 4, 6, 2, 8};
    int n1 = sizeof(arr1) / sizeof(arr1[0]);
    int result1[n1];
    find_previous_greater_element(arr1, n1, result1);

    printf("Input Array: [15, 10, 18, 12, 4, 6, 2, 8]\n");
    printf("PGE Array:   ");
    for (int i = 0; i < n1; i++) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
000gn/T/"tempCodeRunnerFile
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Input Array: [15, 10, 18, 12, 4, 6, 2, 8]
PGE Array:   -1, 15, -1, 18, 12, 12, 6, 12

Input Array: [1, 2, 3, 4, 5]
PGE Array:   -1, -1, -1, -1, -1
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

void productExceptSelf(int* nums, int n, int* answer) {
    if (n == 0) return;

    long long prefix_product = 1;

    for (int i = 0; i < n; i++) {
        answer[i] = (int)prefix_product;
        prefix_product *= nums[i];
    }

    long long suffix_product = 1;

    for (int i = n - 1; i >= 0; i--) {
        answer[i] *= (int)suffix_product;
        suffix_product *= nums[i];
    }
}

int main() {
    int nums1[] = {1, 2, 3, 4};
    int n1 = sizeof(nums1) / sizeof(nums1[0]);
    int answer1[n1];

    productExceptSelf(nums1, n1, answer1);

    printf("Input: [1, 2, 3, 4]\n");
    printf("Output: [");
    for (int i = 0; i < n1; i++) {
        printf("%d%s", answer1[i], (i == n1 - 1 ? "" : ", "));
    }
    printf("]\n");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Input: [1, 2, 3, 4]
Output: [24, 12, 8, 6]

Input: [-1, 1, 0, -3, 3]
Output: [0, 0, 9, 0, 0]
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <limits.h>

int max_subarray_sum_k(int arr[], int n, int k) {
    if (n == 0 || k <= 0 || k > n) {
        return 0;
    }

    int max_sum = INT_MIN;
    int current_window_sum = 0;

    for (int i = 0; i < n; i++) {
        current_window_sum += arr[i];

        if (i >= k - 1) {
            if (current_window_sum > max_sum) {
                max_sum = current_window_sum;
            }

            current_window_sum -= arr[i - (k - 1)];
        }
    }

    return max_sum;
}

int main() {
    int arr1[] = {1, 4, 2, 10, 2, 3, 1, 0, 20};
    int n1 = sizeof(arr1) / sizeof(arr1[0]);
    int k1 = 4;

    printf("Array: [1, 4, 2, 10, 2, 3, 1, 0, 20], k = %d\n", k1);
    printf("Maximum subarray sum of size %d is: %d\n", k1, max_subarray_sum_k(arr1, n1, k1));

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
000gn/T/"tempCodeRunnerFile
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array: [1, 4, 2, 10, 2, 3, 1, 0, 20], k = 4
Maximum subarray sum of size 4 is: 24

Array: [10, -5, 20, 15, -10, 5], k = 3
Maximum subarray sum of size 3 is: 30
nikhilsisodia@Nikhils-MacBook-Air T %
```

```c
#include <stdio.h>
#include <stdlib.h>

void print_max_of_subarrays(int arr[], int n, int k) {
    if (n == 0 || k <= 0 || k > n) {
        printf("\n");
        return;
    }

    // Deque stores indices of array elements.
    // Size n is sufficient for the deque array.
    int* deque = (int*)malloc(n * sizeof(int));
    int front = 0;
    int rear = -1;

    // --- 1. Process the First Window (i = 0 to k-1) ---
    for (int i = 0; i < k; i++) {
        // Remove smaller elements from the rear (maintaining monotonic decrease)
        while (front <= rear && arr[i] >= arr[deque[rear]]) {
            rear--;
        }
        // Add current element's index to the rear
        rear++;
        deque[rear] = i;
    }

    // --- 2. Process Remaining Windows (i = k to n-1) ---
    for (int i = k; i < n; i++) {
        // The element at the front of the deque is the maximum of the previous window (ending at i-1)
        printf("%d ", arr[deque[front]]);

        // Remove the index from the front if it's outside the current window
        if (deque[front] == i - k) {
            front++;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/va
000gn/T/"tempCodeRunnerFile
nikhilsisodia@Nikhils-MacBook-Air ~ % cd "/var/folders/mp/q_89ln6x0_1_3z1sjbnr0mnr0000gn/T/" && gcc tempCodeRunn
empCodeRunnerFile
Array: [1, 3, -1, -3, 5, 3, 6, 7], k = 3
Maximums: 3 3 5 5 6 7

Array: [10, 5, 2, 7, 8, 7], k = 3
Maximums: 10 7 8 8
nikhilsisodia@Nikhils-MacBook-Air T %
```