Riphah International University Of Lahore

Name:

Hanzla Ahmad

SAP ID:

61002

Subject:

Data Structure & Algorithms

Submitted to:

Mrs. Atiqa Islam

Lab 4

Task 1

1- Find First and Last Occurrence:

- **Problem:** Given an array, find the first and last occurrences of a given target value.
- **Example:** Input: arr = [2, 4, 2, 6, 2, 3, 4], target = 2

  Output: First occurrence = 0, Last occurrence = 4

- **Hint:** Use linear search to iterate through the array, updating the positions when the target is found.

Code:

```cpp
DSA > Tasks > Lab 4 > C++ task1.cpp > ⦿ main()
1    #include <iostream>
2    using namespace std;
3
4    int main() {
5        int arr[] = {2, 4, 2, 6, 2, 3, 4};
6        int n = sizeof(arr) / sizeof(arr[0]);
7        int target = 2;
8
9        int first = -1;
10       int last = -1;
11
12       for (int i = 0; i < n; i++) {
13           if (arr[i] == target) {
14               if (first == -1) {
15                   first = i;
16               }
17               last = i;
18           }
19       }
20
21       if (first != -1) {
22           cout << "First occurrence = " << first << endl;
23           cout << "Last occurrence = " << last << endl;
24       } else {
25           cout << "Target not found!" << endl;
26       }
27
28       return 0;
29   }
```

Output:

```
$ g++ task1.cpp -o task1 && ./task1.exe
First occurrence = 0
Last occurrence = 4
```

Task 2

2- Count Occurrences of a Number:

- **Problem:** Given an array, count how many times a given target number appears.
- **Example:** Input: arr = [1, 2, 2, 3, 2, 5], target = 2

Output: 3

Code:

```cpp
#include <iostream>
using namespace std;

int main() {
    int arr[] = {1, 2, 2, 3, 2, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int target = 2;

    int count = 0;

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

    cout << "The number " << target << " appears " << count << " times." << endl;

    return 0;
}
```

Output:

```
$ g++ task2.cpp -o task2 && ./task2.exe
The number 2 appears 3 times.
```

Task 3

3- Check for Pair with Given Sum:

- **Problem:** Given an array and a target sum, determine if any two numbers add up to the target.
- **Example:** Input: arr = [3, 1, 4, 6, 5], target_sum = 9

  Output: True (because 4 + 5 = 9)

- **Hint:** Use nested loops or a hash table to store elements as you search.

Code: (Used Hash)

```cpp
DSA > Tasks > Lab 4 > C++ task3.cpp > ⊗ main()
1    #include <iostream>
2    #include <unordered_set>
3    using namespace std;
4
5    //Using hash
6    int main() {
7        int arr[] = {3, 1, 4, 6, 5};
8        int n = sizeof(arr) / sizeof(arr[0]);
9        int target = 9;
10
11       unordered_set<int> seen;
12       bool found = false;
13
14       for (int i = 0; i < n; i++) {
15           int complement = target - arr[i];
16
17           if (seen.find(complement) != seen.end()) {
18               cout << "Pair found: " << arr[i] << " + " << complement << " = " << target << endl;
19               found = true;
20               break;
21           }
22
23           seen.insert(arr[i]);
24       }
25
26       if (!found) {
27           cout << "No pair found with the given sum." << endl;
28       }
29
30       return 0;
31   }
```

Output:

```
$ g++ task3.cpp -o task3 && ./task3.exe
Pair found: 6 + 3 = 9
```