Experiment 2

Student Name: Ankita **UID:** 22BCS17162

Branch: CSE **Section/Group:**IOT_NTPP_602-A

Semester: 6th **Date of Performance:**20-01-25

Subject Name: AP- 2 **Subject Code:** 22CSP-351

Aim:

a) Two sum problem
b) Remove Duplicates from Sorted Array
c) Jump Game

Objective: To learn about arrays

Code:

a)

```
#include <vector>
#include <unordered_map>
class Solution {
public: std::vector<int> twoSum(const std::vector<int>& nums, int target) {
std::unordered_map<int, int> numMap;
for (int i = 0; i < nums.size(); ++i)
{ int complement = target - nums[i];
if (numMap.find(complement) != numMap.end())
{ return {numMap[complement], i};
}
 numMap[nums[i]] = i;
}
return {};
}
};
```

b)

```
#include <vector>
using namespace std;
class Solution {
public:
int removeDuplicates(vector<int>& nums) {
if(nums.empty())
return 0;
int k=1;
for(int i=1;i<nums.size();i++) {
if(nums[i]!=nums[i-1]) {
nums[k]=nums[i];
k++;
}}
return k;
}};
```

c)

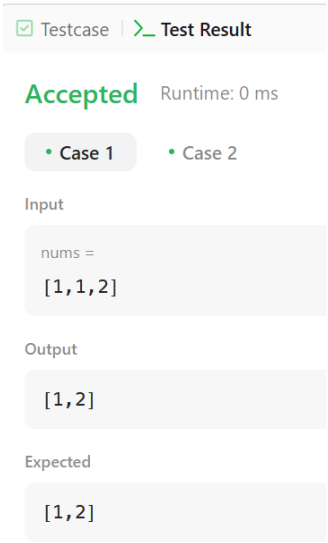
```
#include <vector>
using namespace std;
class Solution {
public:
bool canJump(vector<int>& nums) {
int n = nums.size();
int reachable = 0;
for (int i = 0; i < n; i++) {
if (i > reachable) {
return false;
}
reachable = max(reachable, i + nums[i]);
if (reachable >= n - 1) {
return true;
}
}
return false;
}
};
```

Output:

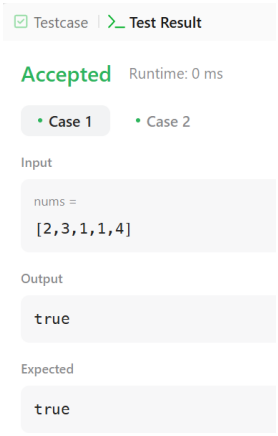
a)


Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[2,7,11,15]
target =
9
Output
[0,1]
Expected
[0,1]

b)


Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[1,1,2]
Output
[1,2]
Expected
[1,2]

c)



Learning Outcomes:

a) Learn to efficiently find pairs of numbers in an array that sum to a target using a hashmap.
b) Understand the concept of complementing values to optimize search operations in arrays.
c) Gain proficiency in handling edge cases and constraints in algorithmic problems.
d) Master the implementation of a FIFO queue using two stacks, ensuring correct operation of queue functions.