Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring , Year:2024), B.Sc. in CSE (Day)

CLP #04

Course Title: Artificial Intelligence Lab

Course Code: CSE - 316 Section: 213_D5

Lab Experiment Name: Introduction to Basic Operations on Python.

Student Details

Name	ID
MD Dulal Hossain	213902116

Submission Date : 19– 03 - 2023

Course Teacher's Name : Sagufta Sabah Nakshi

[For Teachers use only: Don't Write Anything inside this box]

CLP Status

Marks: …… Signature:.....

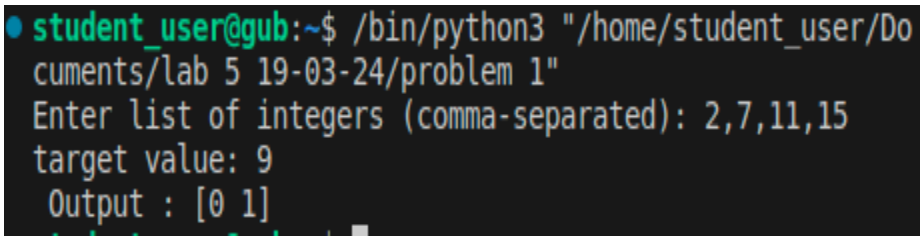
Comments:..... Date:.....

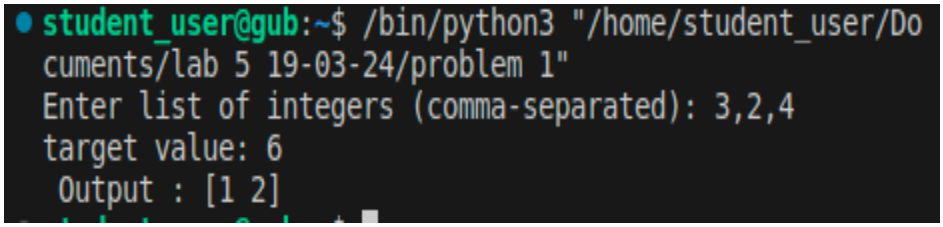
1. PROBLEM 1

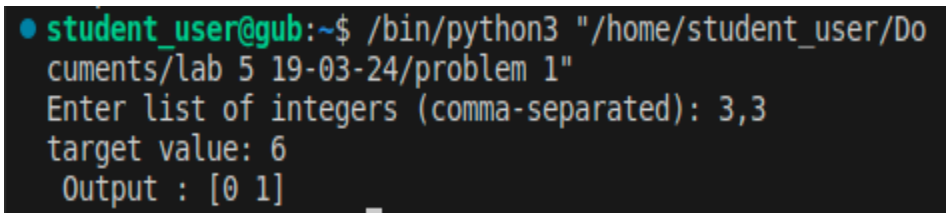
CODE :

```
def two_sum(nums, target):
  seen = {}
  for i, num in enumerate(nums):
    complement = target - num
    if complement in seen:
      return [seen[complement], i]
    seen[num] = i
  return []
num_list_str = input("Enter list of integers (comma-separated): ")
num_list = [int(num) for num in num_list_str.split(",")]
target_str = input("target value: ")
target = int(target_str)
result = two_sum(num_list, target)
if result:
  print(f" Output : [{result[0]} {result[1]}]")
else:
  print(f"No two numbers in the list add up to {target}")
```

OUTPUT :





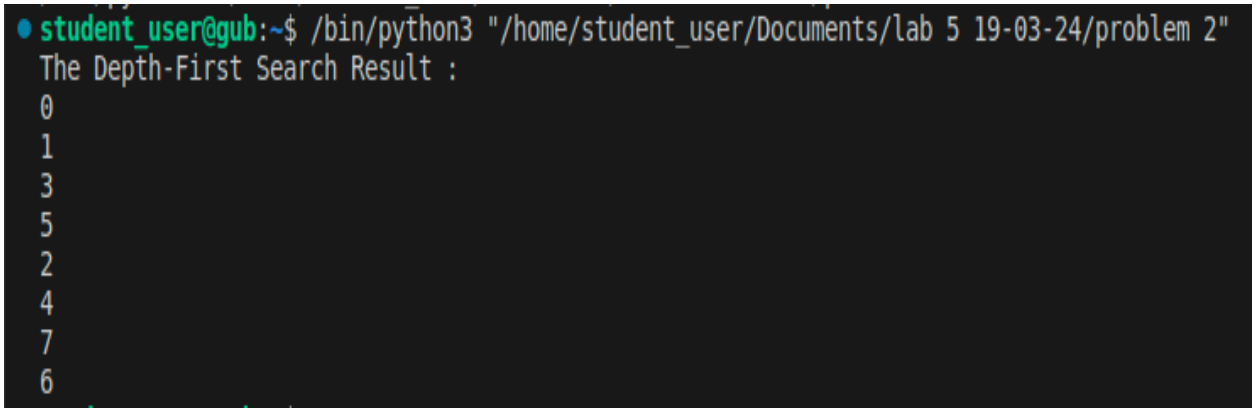


2. PROBLEM 2

CODE :

```
graph = {
    '0': ['1', '2', '3'],
    '1': ['3'],
    '2': ['4'],
    '3': ['5','6'],
    '4': ['5', '7'],
    '5': ['2'],
    '6': [],
    '7': []

}
visited = set()

def dfs(visited, graph, node):
    if node not in visited:
        print (node)
        visited.add(node)
        for neighbour in graph[node]:
            dfs(visited, graph, neighbour)


print("The Depth-First Search Result : ")
dfs(visited, graph,'0')
```

OUTPUT :

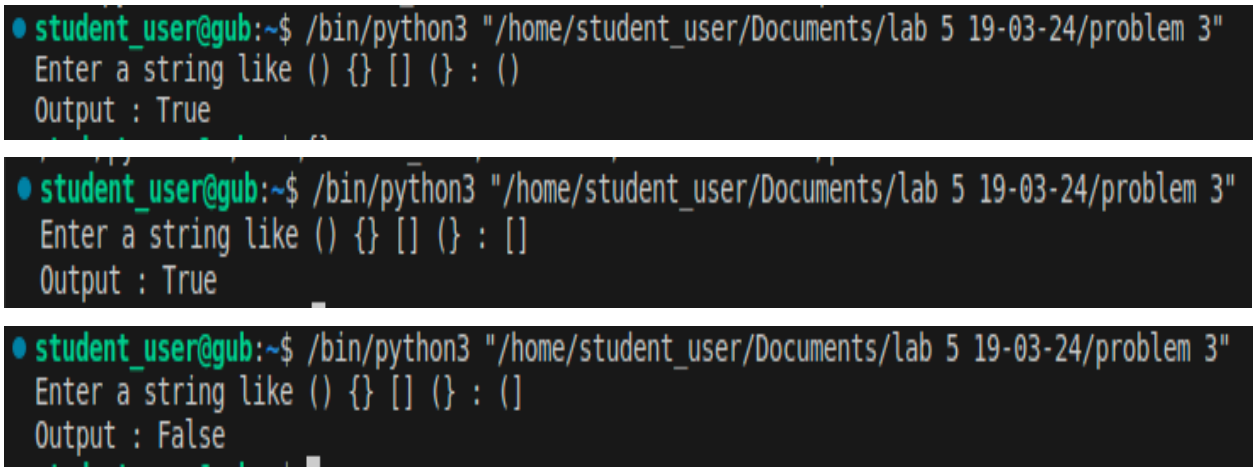


3. PROBLEM 3

CODE :

```
def is_valid(s):

  stack = []
  mapping = {")": "(", "}": "{", "]": "["}

  for char in s:
    if char in mapping:
      top = stack.pop() if stack else '#'
      if top != mapping[char]:
        return False
    else:
      stack.append(char)

  return len(stack) == 0


user_string = input("Enter a string like () {} [] (} : ")


result = is_valid(user_string)

print("Output :", result)
```

OUTPUT :

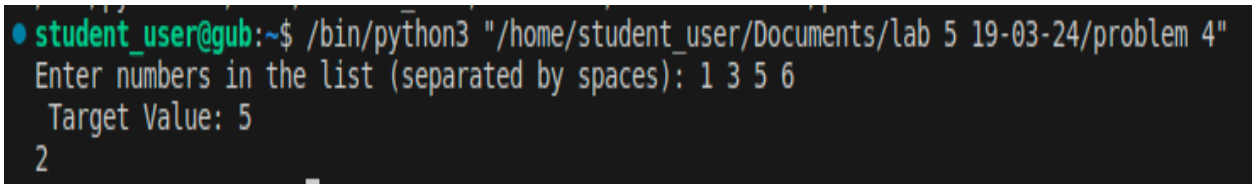


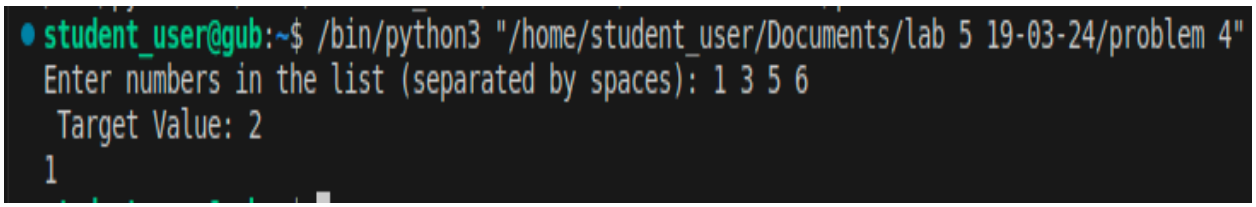
4. PROBLEM 4

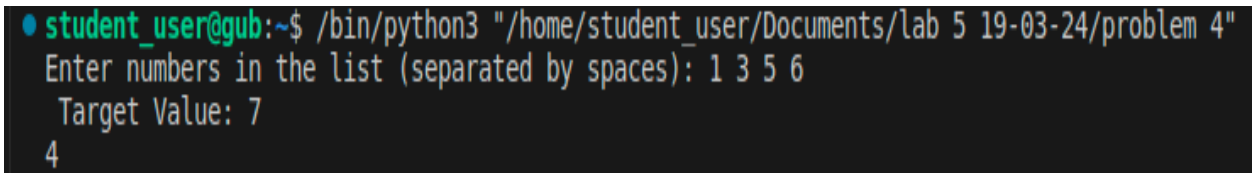
CODE :

```
def search_insert_position(nums, target):
    left, right = 0, len(nums) - 1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            return mid
        elif nums[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return left
nums_input = input("Enter numbers in the list (separated by spaces): ")
nums = list(map(int, nums_input.split()))

target = int(input(" Target Value: "))

print(search_insert_position(nums, target))
```

OUTPUT :


student_user@gub:~$ /bin/python3 "/home/student_user/Documents/lab 5 19-03-24/problem 4"
Enter numbers in the list (separated by spaces): 1 3 5 6
Target Value: 5
2


student_user@gub:~$ /bin/python3 "/home/student_user/Documents/lab 5 19-03-24/problem 4"
Enter numbers in the list (separated by spaces): 1 3 5 6
Target Value: 2
1


student_user@gub:~$ /bin/python3 "/home/student_user/Documents/lab 5 19-03-24/problem 4"
Enter numbers in the list (separated by spaces): 1 3 5 6
Target Value: 7
4