CSE 352 - LAB 1

Arnav Jain - 220002018

Objective Questions:

Question 1 - A) 3, 1

Question 2 - D) All of the above

Question 3 - C) True and False are valid Boolean literals.

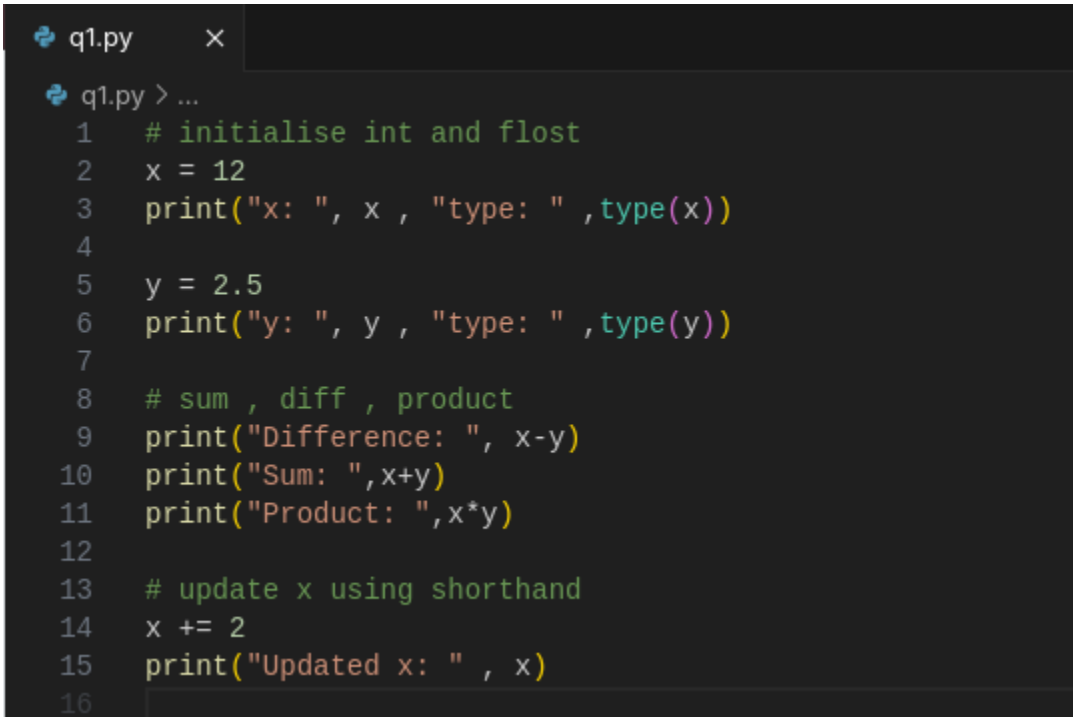
Question 4 - A) [2, 3, 4]

Question 5 - A) set()

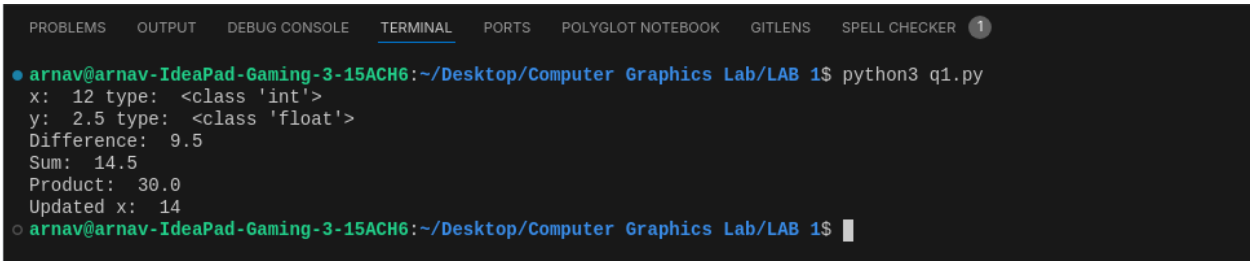
Coding Problems:

Question 1

Code:

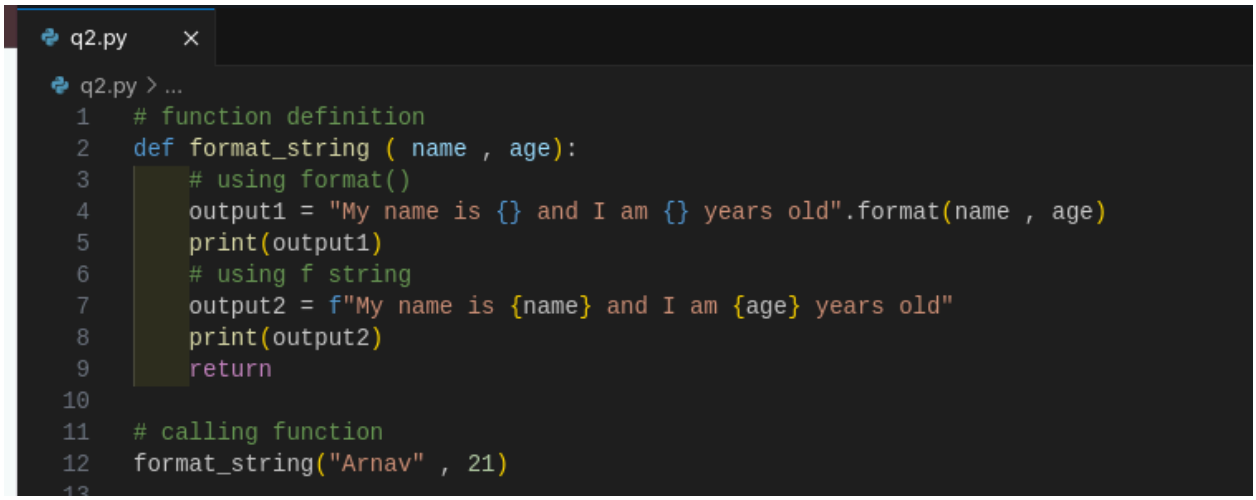


Result:

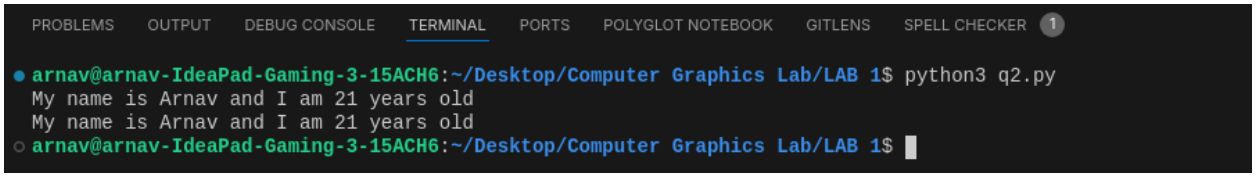


Question 2

Code:



Result:



Question 3

Code:

q3.py

```python
# Create a list of numbers from 1 to 10
numbers = list(range(1, 11))

# Print the square of each number using list comprehension
squares = [x**2 for x in numbers]
print("Squares of numbers:", squares)

# Extract all even numbers from the list using slicing
even_numbers = numbers[1::2]
print("Even numbers:", even_numbers)
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$ python3 q3.py
Squares of numbers: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
Even numbers: [2, 4, 6, 8, 10]
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$
```

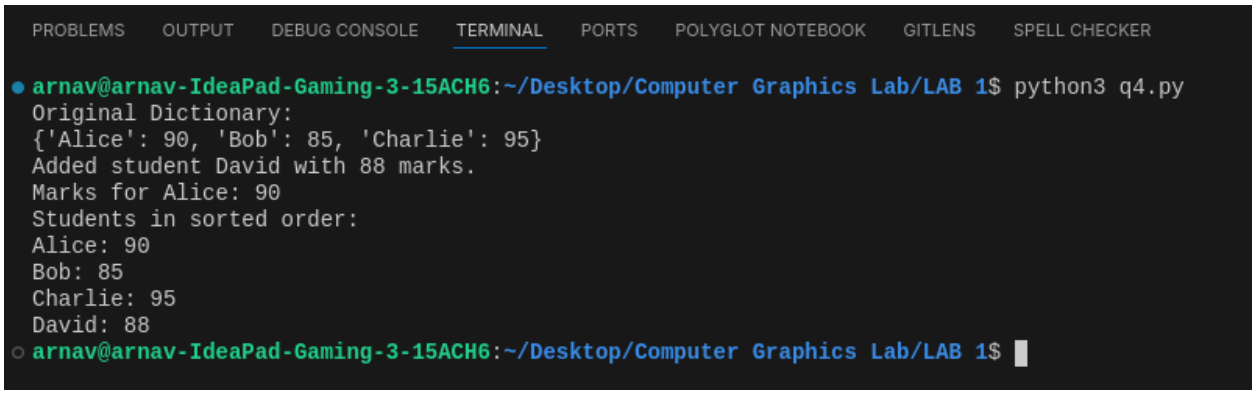
Question 4

Code:

q4.py

```python
# Creates a dictionary to store student names and their marks.
def create_student_dictionary():
    students = {
        "Alice": 90,
        "Bob": 85,
        "Charlie": 95
    }
    return students

# Adds a new student to the dictionary.
def add_student(students, name, marks):
    students[name] = marks
    print(f"Added student {name} with {marks} marks.")

# Retrieves the marks of a specific student.
def get_student_marks(students, name):
    if name in students:
        return students[name]
    else:
        return "Student not found."

# Prints the dictionary in sorted order of names.
def print_sorted_students(students):
    for name in sorted(students):
        print(f"{name}: {students[name]}")

# Create the initial dictionary
student_dict = create_student_dictionary()
print("Original Dictionary:")
print(student_dict)

# Add a new student
add_student(student_dict, "David", 88)

# Get marks of a student
student_name = "Alice"
marks = get_student_marks(student_dict, student_name)
print(f"Marks for {student_name}: {marks}")

# Print students in sorted order
print("Students in sorted order:")
print_sorted_students(student_dict)
```

Result:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
POLYGLOT NOTEBOOK
GITLENS
SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$ python3 q4.py
Original Dictionary:
{'Alice': 90, 'Bob': 85, 'Charlie': 95}
Added student David with 88 marks.
Marks for Alice: 90
Students in sorted order:
Alice: 90
Bob: 85
Charlie: 95
David: 88
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$

Question 5

Code:

```python
# Create two sets of integers
set1 = set()
set2 = set()

for x in range(1 , 7):
    set1.add(x)

for x in range(6 , 9):
    set2.add(x)

# Perform set operations
union_set = set1.union(set2)
intersection_set = set1.intersection(set2)
difference_set1 = set1.difference(set2)
difference_set2 = set2.difference(set1)

# Print results
print("Set 1:", set1)
print("Set 2:", set2)
print("Union:", union_set)
print("Intersection:", intersection_set)
print("Difference of Set 1 from Set 2:", difference_set1)
print("Difference of Set 2 from Set 1:", difference_set2)

# Find unique elements from both sets
unique_elements = list(set1.symmetric_difference(set2))
print("Unique elements:", unique_elements)
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$ python3 q5.py
Set 1: {1, 2, 3, 4, 5, 6}
Set 2: {8, 6, 7}
Union: {1, 2, 3, 4, 5, 6, 7, 8}
Intersection: {6}
Difference of Set 1 from Set 2: {1, 2, 3, 4, 5}
Difference of Set 2 from Set 1: {8, 7}
Unique elements: [1, 2, 3, 4, 5, 7, 8]
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$
```

Question 6

Code:

```python
def tuple_operations(input_tuple):
    if not input_tuple:
        return None, None, None  # Handle empty tuple

    max_value = max(input_tuple)
    min_value = min(input_tuple)
    sum_value = sum(input_tuple)

    return max_value, min_value, sum_value

# Example usage:
my_tuple = (10, 5, 20, 3, 8)
max_val, min_val, sum_val = tuple_operations(my_tuple)

print(f"Maximum: {max_val}")
print(f"Minimum: {min_val}")
print(f"Sum: {sum_val}")
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$ python3 q6.py
Maximum: 20
Minimum: 3
Sum: 46
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computer Graphics Lab/LAB 1$
```

For code files , refer here:

https://github.com/arnavjain2710/Computer-Graphics-Lab/tree/main/LAB%201