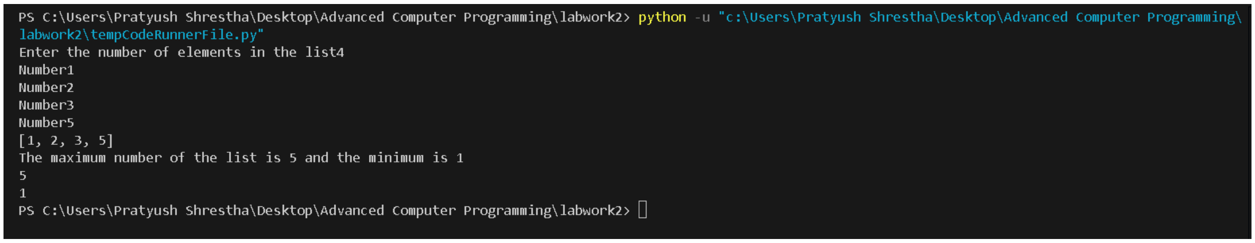
1. Write a program to input n numbers and store them in a list. Then perform the following operations: i) Using built in functions ii) without using built-in functions: a. Find the maximum and minimum number b. Sort the list in ascending order c. Remove duplicate elements

a. Find the maximum and minimum number

```
#Write a program to input n numbers and store them in a list.Find the maximum and
minimum number
n = int(input("Enter the number of elements in the list"))
list1 = []
for i in range(n):
    a = int(input("Number"))
    list1.append(a)
print(list1)

#Using built-in functions
print("The maximum number of the list is", max(list1) ,"and the minimum is",
min(list1))

#Without using built-in functions
max = list1[0]
min = list1[0]
for element in list1:
    if element>max:
        max = element
    if element<min:
        min = element
print(max)
print(min)
```

Screenshot:



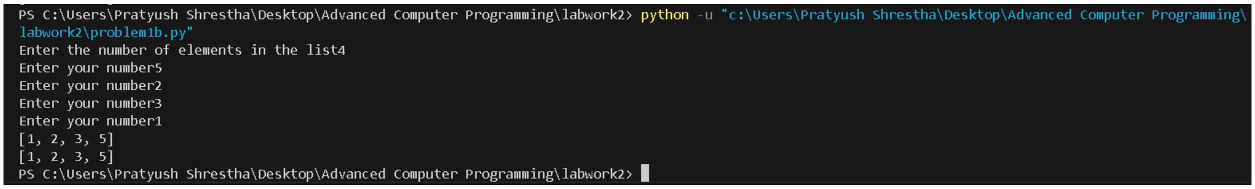
```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
Enter the number of elements in the list4
Number1
Number2
Number3
Number5
[1, 2, 3, 5]
The maximum number of the list is 5 and the minimum is 1
5
1
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

b. Sort the list in ascending order

```
#Sort the list in ascending order
list1 = []
```

```python
n = int(input("Enter the number of elements in the list"))
for i in range(n) :
    a = int(input("Enter your number"))
    list1.append(a)
#Using built-in functions
print(sorted(list1))
#Without using built-in functions
for i in range(len(list1)-1):
    for j in range(len(list1) - i - 1):
        if list1[j] < list1[j+1]:
            list1[j], list1[j + 1] = list1[j + 1], list1[j]

print(list1)
```

Screenshot:

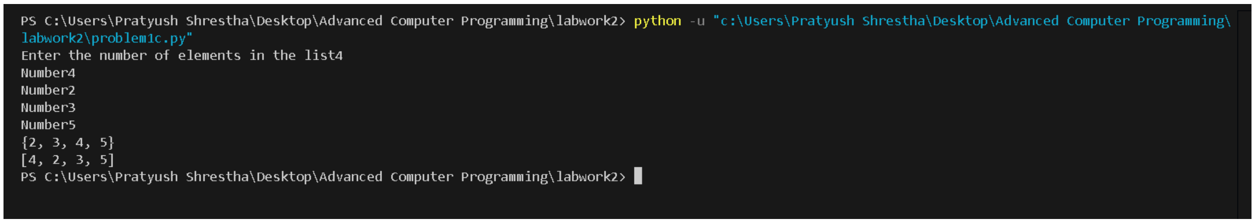


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\problem1b.py"
Enter the number of elements in the list4
Enter your number5
Enter your number2
Enter your number3
Enter your number1
[1, 2, 3, 5]
[1, 2, 3, 5]
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

c. Remove duplicate elements

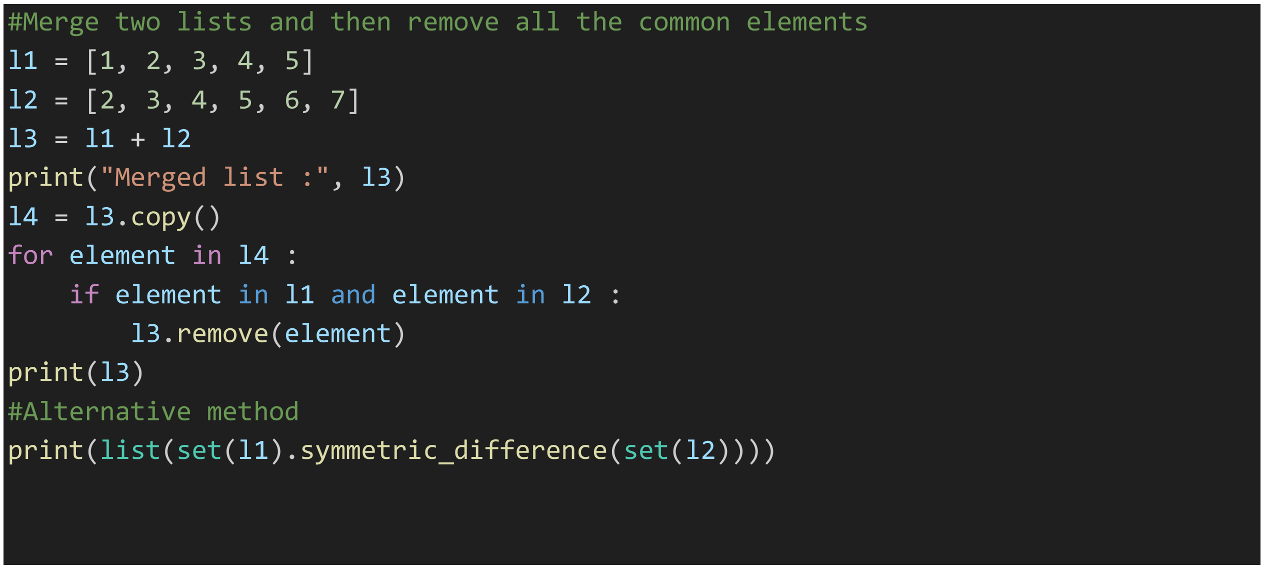
```python
#Remove duplicate elements
n = int(input("Enter the number of elements in the list"))
list1 = []
for i in range(n):
    a = int(input("Number"))
    list1.append(a)

#Using built-in methods
print(set(list1))

l2 = []
#Without using built-in methods
for element in list1:
    if element not in l2 :
        l2.append(element)
print(l2)
```

Screenshot:



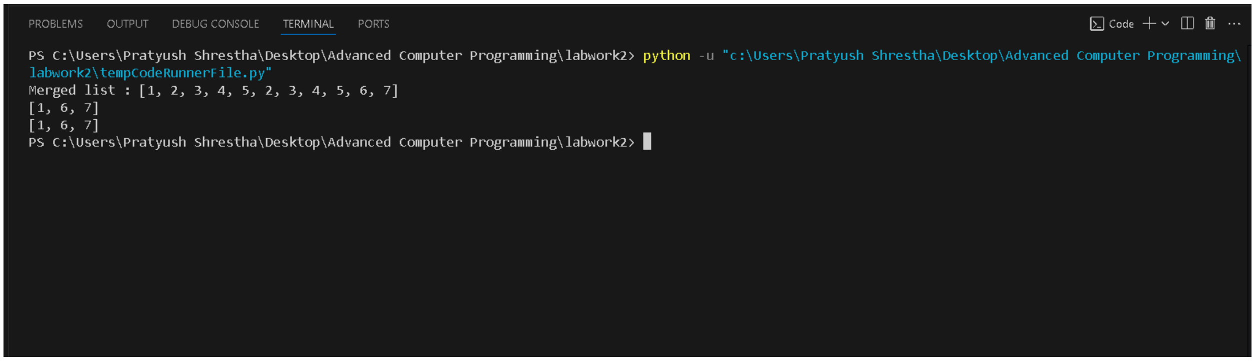
```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\problem1c.py"
Enter the number of elements in the list4
Number4
Number2
Number3
Number5
{2, 3, 4, 5}
[4, 2, 3, 5]
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

2. Given two lists of integers, write a program to merge them into a single list and then remove the elements that are common in both.



```
#Merge two lists and then remove all the common elements
l1 = [1, 2, 3, 4, 5]
l2 = [2, 3, 4, 5, 6, 7]
l3 = l1 + l2
print("Merged list :", l3)
l4 = l3.copy()
for element in l4 :
    if element in l1 and element in l2 :
        l3.remove(element)
print(l3)
#Alternative method
print(list(set(l1).symmetric_difference(set(l2))))
```

Screenshot:

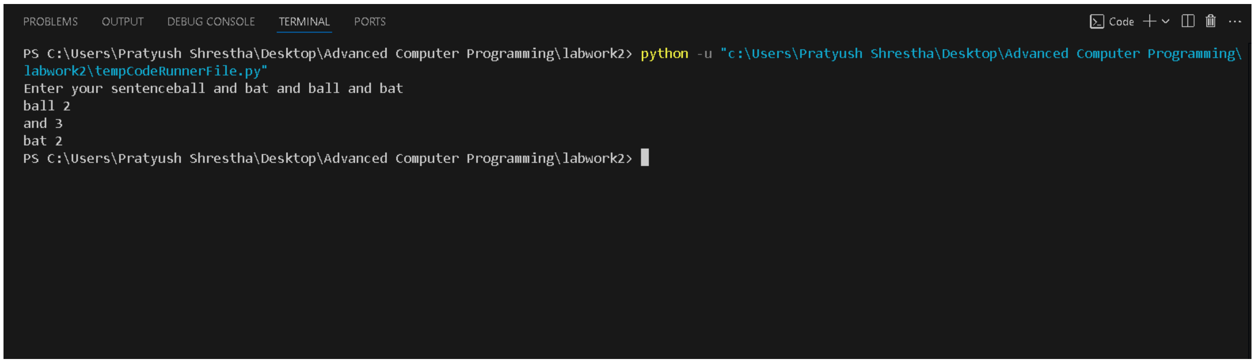


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
Merged list : [1, 2, 3, 4, 5, 2, 3, 4, 5, 6, 7]
[1, 6, 7]
[1, 6, 7]
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

3. Create a program that reads a sentence from the user and stores each word as an element of a list. Then count the frequency of each word using only lists.

```python
#Make a list of the words in the sentence and then count the frequency of the
words
sentence = input("Enter your sentence")
words = sentence.split()
unique_word = []
for word in words :
    if word not in unique_word:
        unique_word.append(word)
for word in unique_word:
    print(word, words.count(word))
```

Screenshot :

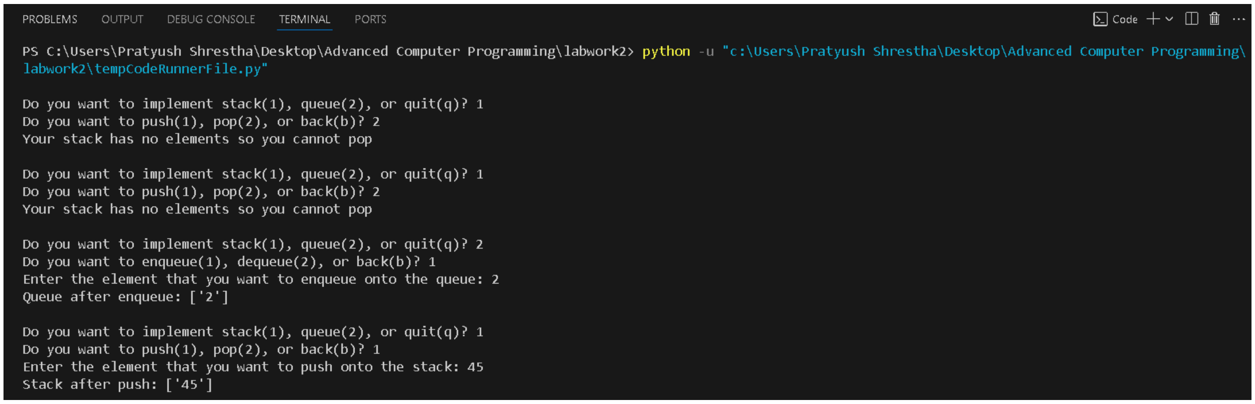


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
Enter your sentenceball and bat and ball and bat
ball 2
and 3
bat 2
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

4. Write a program to simulate a basic stack and queue using a list. Provide options to push, pop (stack), enqueue, and dequeue (queue).

```python
stack = []
queue = []

while True:
    choice = input("\nDo you want to implement stack(1), queue(2), or quit(q)? ")
    if choice == "1":
        schoice = input("Do you want to push(1), pop(2), or back(b)? ")
        if schoice == "1":
            push = input("Enter the element that you want to push onto the stack:
")
            stack.append(push)
            print("Stack after push:", stack)
        elif schoice == "2":
            if len(stack) > 0:
                popped = stack.pop()
                print("Popped element:", popped)
                print("Stack after pop:", stack)
            else:
                print("Your stack has no elements so you cannot pop")
        elif schoice.lower() == "b":
            continue
        else:
            print("Invalid choice for stack operation.")

    elif choice == "2":
        qchoice = input("Do you want to enqueue(1), dequeue(2), or back(b)? ")
        if qchoice == "1":
            enqueue = input("Enter the element that you want to enqueue onto the
queue: ")
            queue.append(enqueue)
            print("Queue after enqueue:", queue)
        elif qchoice == "2":
            if len(queue) > 0:
                dequeued = queue.pop(0)
                print("Dequeued element:", dequeued)
                print("Queue after dequeue:", queue)
            else:
                print("Your queue has no elements so you cannot dequeue")
        elif qchoice.lower() == "b":
            continue
        else:
            print("Invalid choice for queue operation.")

    elif choice.lower() == "q":
        break

    else:
        print("Invalid choice. Please enter 1 for stack, 2 for queue, or q to
quit.")
```

Screenshot:

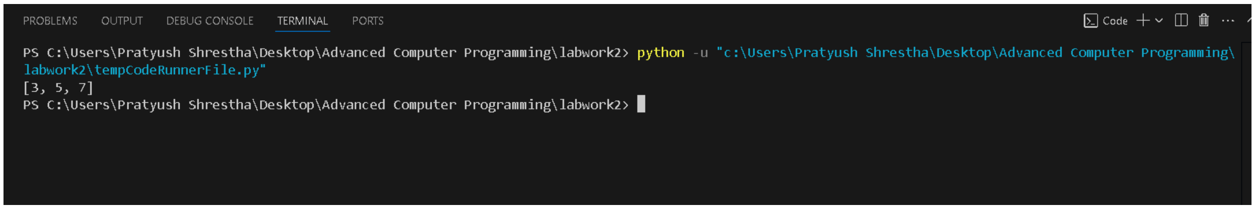


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2\tempCodeRunnerFile.py"

Do you want to implement stack(1), queue(2), or quit(q)? 1
Do you want to push(1), pop(2), or back(b)? 2
Your stack has no elements so you cannot pop

Do you want to implement stack(1), queue(2), or quit(q)? 1
Do you want to push(1), pop(2), or back(b)? 2
Your stack has no elements so you cannot pop

Do you want to implement stack(1), queue(2), or quit(q)? 2
Do you want to enqueue(1), dequeue(2), or back(b)? 1
Enter the element that you want to enqueue onto the queue: 2
Queue after enqueue: ['2']

Do you want to implement stack(1), queue(2), or quit(q)? 1
Do you want to push(1), pop(2), or back(b)? 1
Enter the element that you want to push onto the stack: 45
Stack after push: ['45']
```

5. Write a Python function that accepts a list and returns a new list containing only the elements at even indexes and those that are prime numbers.

```python
l1 = [1, 2, 3, 4, 5, 6, 7]
l2 = []

for element in range(len(l1)):
    if element % 2 == 0 :
        num = l1[element]
        count = 0
        for j in range(2, num):
            if num % j == 0:
                count += 1
        if num > 1 and count == 0:
            l2.append(num)
print(l2)
```

Screenshot:



```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2\tempCodeRunnerFile.py"
[3, 5, 7]
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

6. Write a program to create a tuple of n numbers, then find:

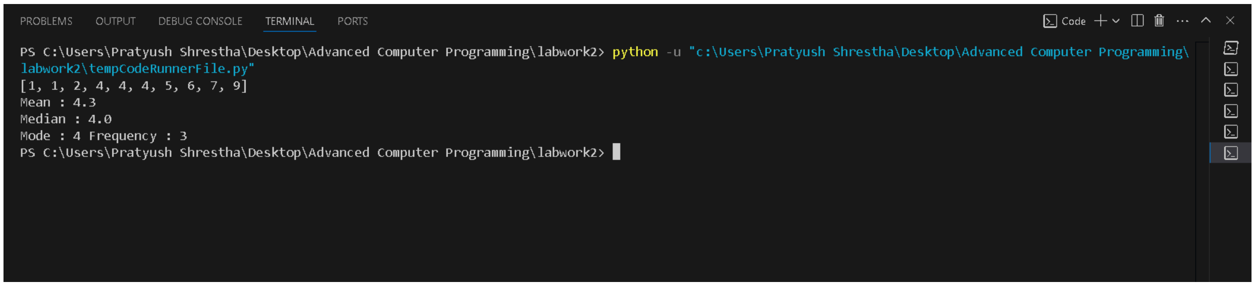
a. The average of the numbers

b. The median

c. The mode (without using libraries)

```python
import random
length = random.choice([10, 11])
list1 = [random.randint(1, 10) for i in range(length)]
list1.sort()
print(list1)
print("Mean :", sum(list1)/len(list1))
if(len(list1) % 2 ==0):
    n1 = list1[(len(list1)//2)-1]
    n2 = list1[(len(list1)//2)]
    print("Median :", (n1+n2)/2)
else:
    print("Median :", list1[len(list1)//2])
unique_list1 = []
for element in list1 :
    if element not in unique_list1:
        unique_list1.append(element)
freq_list = []
for element in unique_list1 :
    freq_list.append(list1.count(element))
if max(freq_list) > 1 :
    print("Mode :", unique_list1[freq_list.index(max(freq_list))], "Frequency :",
max(freq_list))
else :
    print("Mode does not exist")
```

Screenshot:

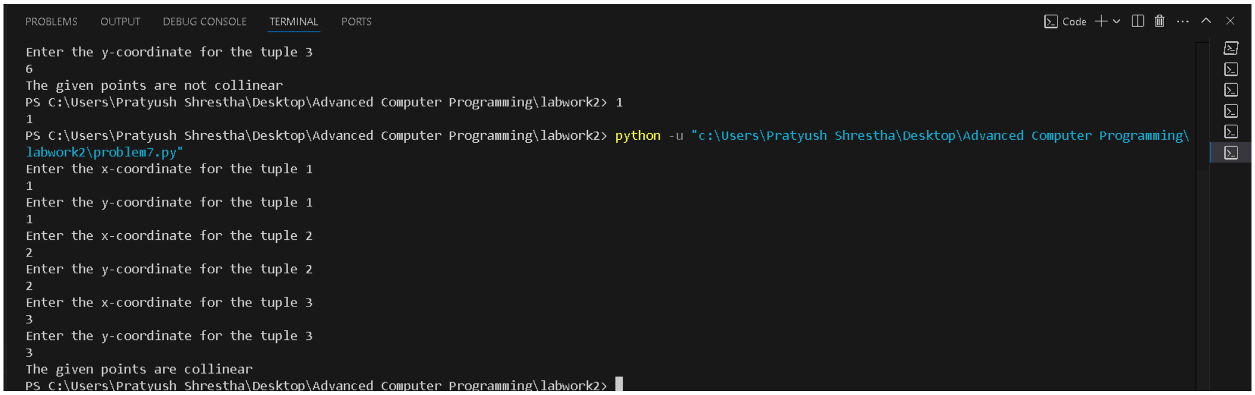


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
[1, 1, 2, 4, 4, 4, 5, 6, 7, 9]
Mean : 4.3
Median : 4.0
Mode : 4 Frequency : 3
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

7. Write a program that receives a list of tuples representing (x, y) coordinates. Determine whether the points form a straight line.

```python
l1 = []
for i in range(3):
    print(f"Enter the x-coordinate for the tuple {i+1}")
    x = int(input())
    print(f"Enter the y-coordinate for the tuple {i+1}")
    y = int(input())
    l1.append((x, y))
if((l1[1][1]-l1[0][1])*(l1[2][0]-l1[0][0])==(l1[1][0]-l1[0][0])*(l1[2][1]-
l1[0][1])):
    print("The given points are collinear")
else:
    print("The given points are not collinear")
```

Screenshot :



```
Enter the y-coordinate for the tuple 3
6
The given points are not collinear
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 1
1
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\problem7.py"
Enter the x-coordinate for the tuple 1
1
Enter the y-coordinate for the tuple 1
1
Enter the x-coordinate for the tuple 2
2
Enter the y-coordinate for the tuple 2
2
Enter the x-coordinate for the tuple 3
3
Enter the y-coordinate for the tuple 3
3
The given points are collinear
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

8. Write a program to input two sets of student roll numbers: one who play cricket and another who play football. Find:

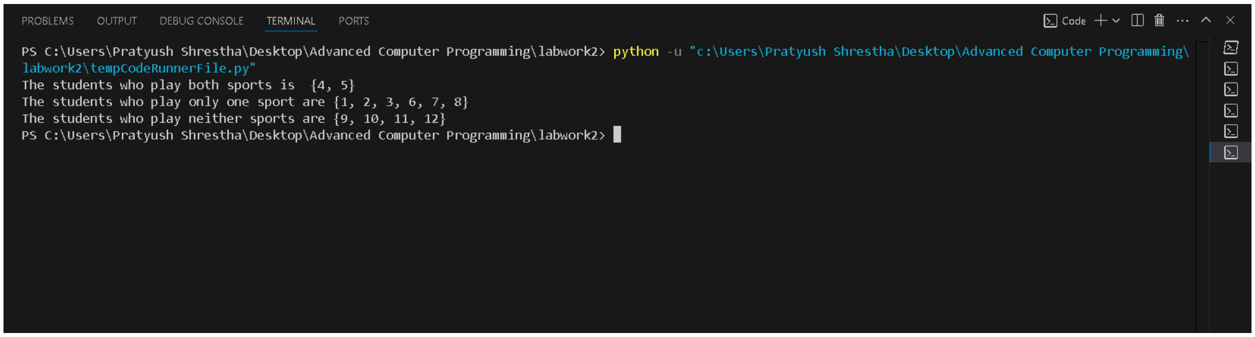
a. Students who play both sports

b. Students who play only one sport

c. Students who play neither (given a master list of all students)

```
C = {1, 2, 3, 4, 5}
F = {4, 5, 6, 7, 8}
S = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
print("The students who play both sports is ", C.intersection(F))
print("The students who play only one sport are", C.symmetric_difference(F))
print("The students who play neither sports are", S.difference((C.union(F))))
```

Screenshot:



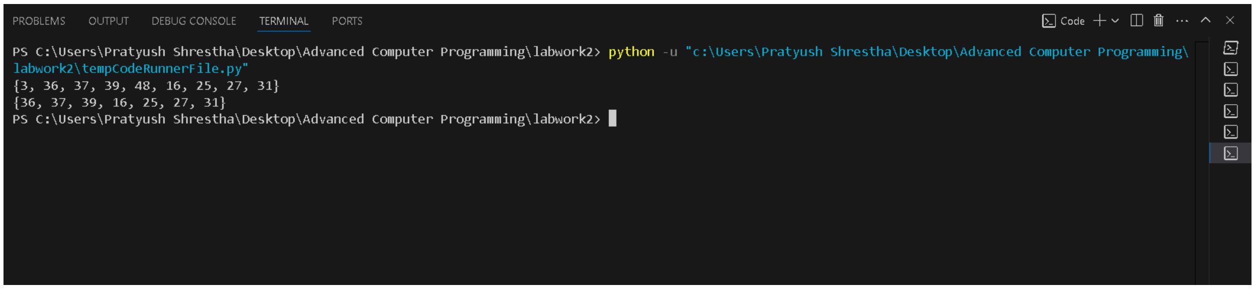
```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
The students who play both sports is  {4, 5}
The students who play only one sport are {1, 2, 3, 6, 7, 8}
The students who play neither sports are {9, 10, 11, 12}
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

9. Create a set of random numbers. Add more numbers until the set has 10 unique elements. Also, remove the smallest and largest element.

```
import random
s = set()
for i in range(10) :
    s.add(random.randint(1, 50))
```

```
print(s)
s.remove(max(s))
s.remove(min(s))
print(s)
```

Screenshot:

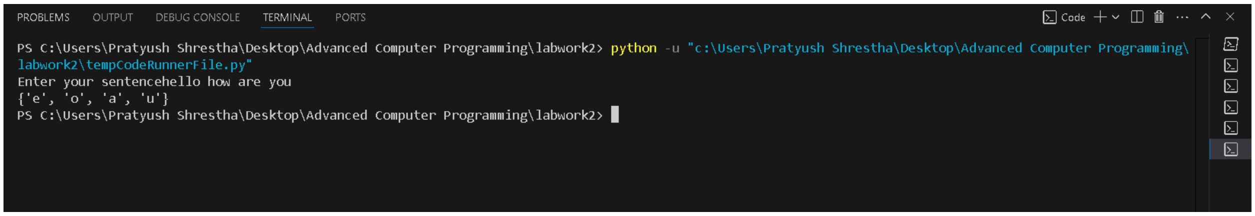


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
{3, 36, 37, 39, 48, 16, 25, 27, 31}
{36, 37, 39, 16, 25, 27, 31}
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

10. Write a Python function that accepts a sentence and returns a set of all unique vowels Used.

```
a = input("Enter your sentence")
a = a.lower()
vowel = []
s = set()
for char in a :
    if char not in vowel :
        vowel.append(char)
for element in vowel :
    if element == "a" or element == "e" or element == "i" or element == "o" or 
element == "u" :
        s.add(element)
print(s)
```

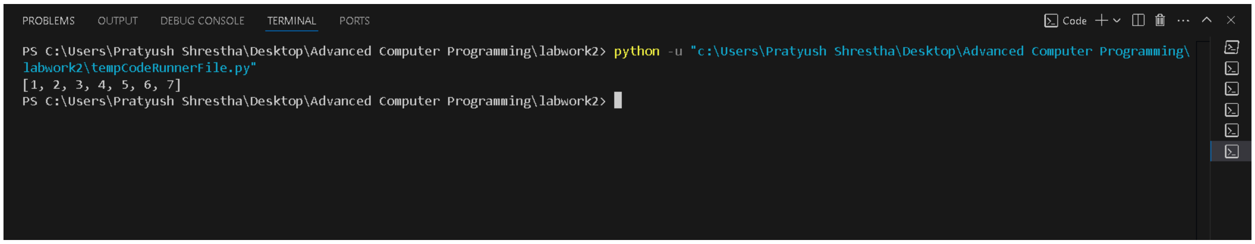
Screenshot:



```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\
labwork2\tempCodeRunnerFile.py"
Enter your sentencehello how are you
{'e', 'o', 'a', 'u'}
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> 
```

11. Given a list of numbers with duplicates, use a set to remove the duplicates. Then, convert it back to a sorted list and display the result.

```
l1 = [1, 2, 3, 4, 5, 5, 5, 6, 6, 7, 1, 2]
l1 = set(l1)
print(sorted(l1))
```

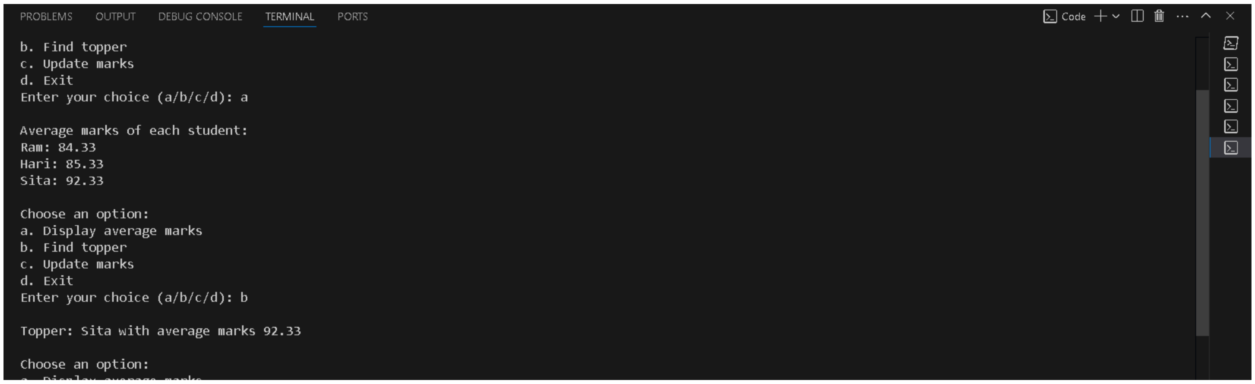


```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2\tempCodeRunnerFile.py"
[1, 2, 3, 4, 5, 6, 7]
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

12. Create a dictionary to store student names as keys and their scores in three subjects as values (in a list). Write functions to: a. Display the average marks of each student b. Find the topper c. Update the marks of a student

```python
students = {
    "Ram": [78, 85, 90],
    "Hari": [88, 76, 92],
    "Sita": [95, 89, 93]
}

while True:
    print("\nChoose an option:")
    print("a. Display average marks")
    print("b. Find topper")
    print("c. Update marks")
    print("d. Exit")
    choice = input("Enter your choice (a/b/c/d): ").lower()

    if choice == 'a':
        print("\nAverage marks of each student:")
        for name in students:
            marks = students[name]
            avg = sum(marks) / len(marks)
            print(f"{name}: {avg:.2f}")

    elif choice == 'b':
        topper = ""
        top_avg = 0
        for name in students:
            marks = students[name]
            avg = sum(marks) / len(marks)
            if avg > top_avg:
                top_avg = avg
                topper = name
        print(f"\nTopper: {topper} with average marks {top_avg:.2f}")
    elif choice == 'c':
        name = input("Enter student name to update: ")
        if name in students:
            try:
                print("Enter new marks for 3 subjects:")
                m1 = int(input("Subject 1: "))
                m2 = int(input("Subject 2: "))
                m3 = int(input("Subject 3: "))
                students[name] = [m1, m2, m3]
                print(f"Updated marks for {name}")
            except ValueError:
                print("Invalid input. Please enter integer marks.")
        else:
            print("Student not found.")

    elif choice == 'd':
        print("Exiting program.")
        break

    else:
        print("Invalid choice.")
```

Screenshot:

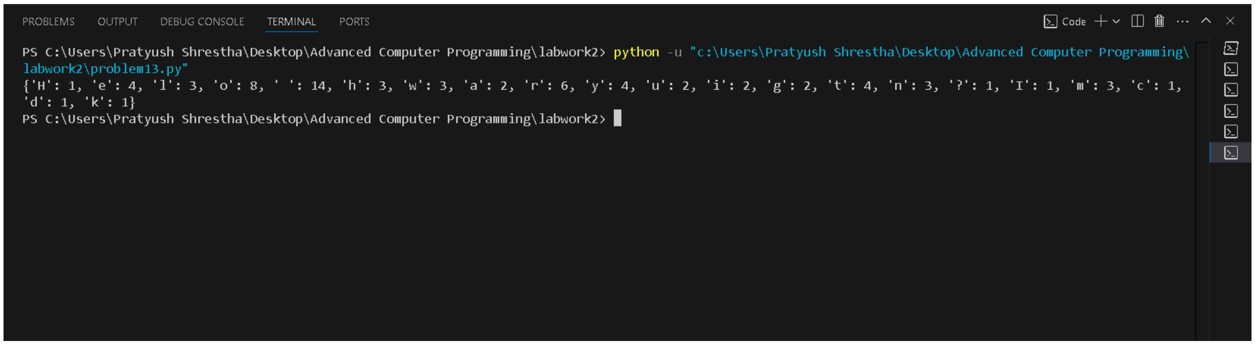


```
b. Find topper
c. Update marks
d. Exit
Enter your choice (a/b/c/d): a

Average marks of each student:
Ram: 84.33
Hari: 85.33
Sita: 92.33

Choose an option:
a. Display average marks
b. Find topper
c. Update marks
d. Exit
Enter your choice (a/b/c/d): b

Topper: Sita with average marks 92.33

Choose an option:
a. Display average marks
```

13. Write a program that reads a text and counts the frequency of each character (excluding spaces and special characters) using a dictionary.

```
a = "Hello how are you right now? I am currently trying to do my homework "
count_dict = {}
for char in a :
```

```python
    if char not in count_dict :
        count_dict.update({char:a.count(char)})
print(count_dict)
```



```
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2> python -u "c:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2\problem13.py"
{'H': 1, 'e': 4, 'l': 3, 'o': 8, ' ': 14, 'h': 3, 'w': 3, 'a': 2, 'r': 6, 'y': 4, 'u': 2, 'i': 2, 'g': 2, 't': 4, 'n': 3, '?': 1, 'I': 1, 'm': 3, 'c': 1,
'd': 1, 'k': 1}
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

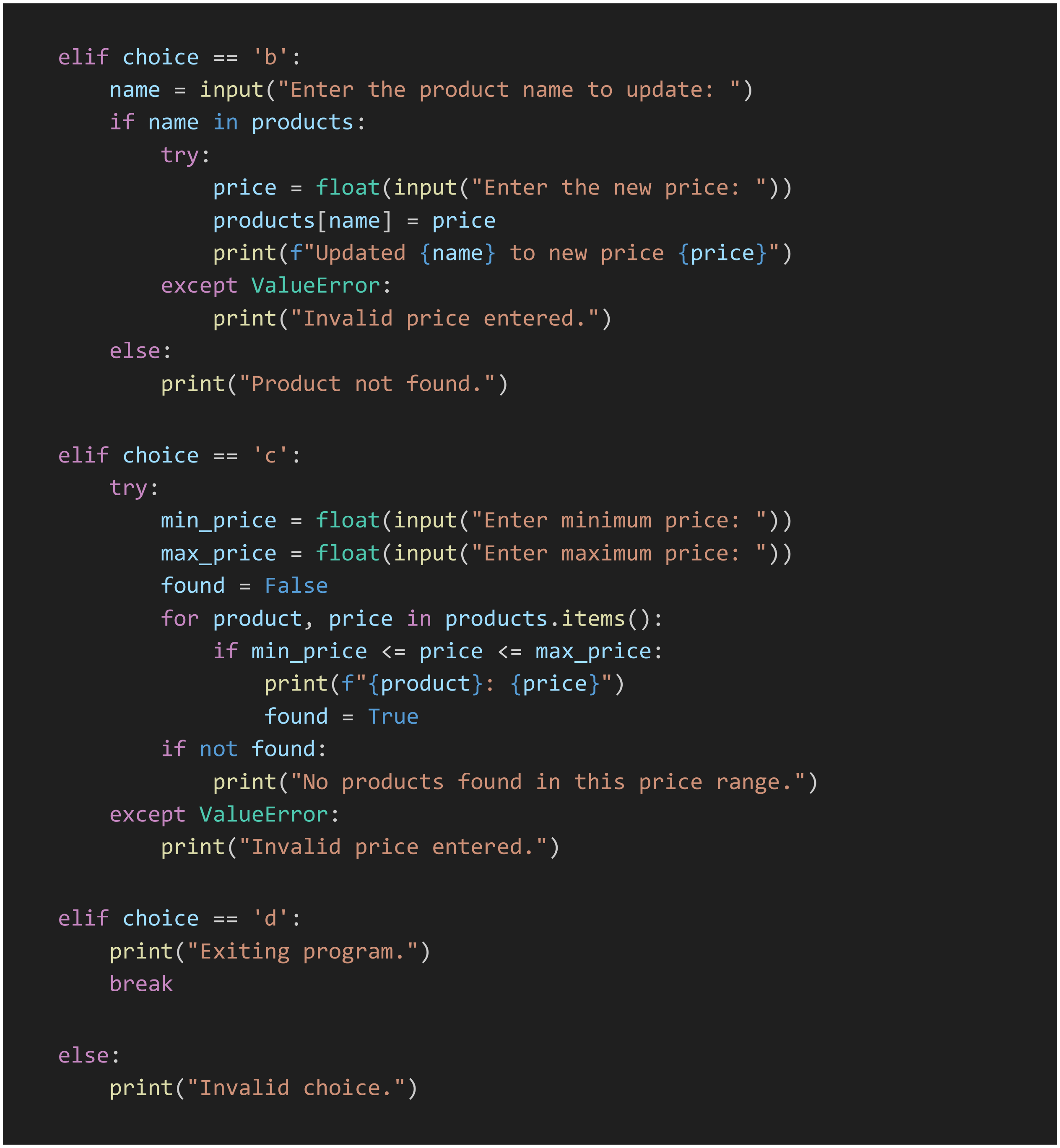
14. Build a dictionary where the keys are product names and the values are their prices. Implement options to:

a. Add a new product

b. Update price of an existing product

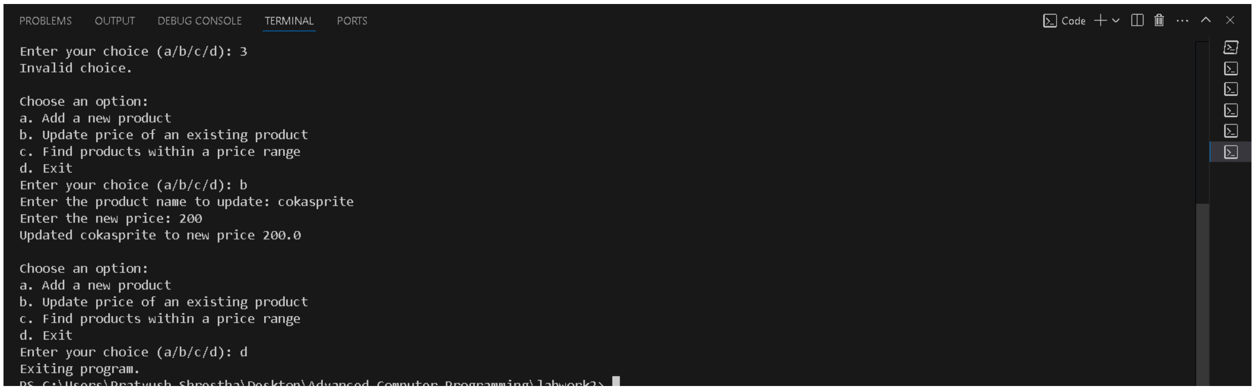
c. Find products within a given price range

```python
products = {}

while True:
    print("\nChoose an option:")
    print("a. Add a new product")
    print("b. Update price of an existing product")
    print("c. Find products within a price range")
    print("d. Exit")
    choice = input("Enter your choice (a/b/c/d): ").lower()

    if choice == 'a':
        name = input("Enter the product name: ")
        if name in products:
            print("Product already exists.")
        else:
            try:
                price = float(input("Enter the price: "))
                products[name] = price
                print(f"Added {name} with price {price}")
            except ValueError:
                print("Invalid price entered.")
```



```
elif choice == 'b':
    name = input("Enter the product name to update: ")
    if name in products:
        try:
            price = float(input("Enter the new price: "))
            products[name] = price
            print(f"Updated {name} to new price {price}")
        except ValueError:
            print("Invalid price entered.")
    else:
        print("Product not found.")

elif choice == 'c':
    try:
        min_price = float(input("Enter minimum price: "))
        max_price = float(input("Enter maximum price: "))
        found = False
        for product, price in products.items():
            if min_price <= price <= max_price:
                print(f"{product}: {price}")
                found = True
        if not found:
            print("No products found in this price range.")
    except ValueError:
        print("Invalid price entered.")

elif choice == 'd':
    print("Exiting program.")
    break

else:
    print("Invalid choice.")
```

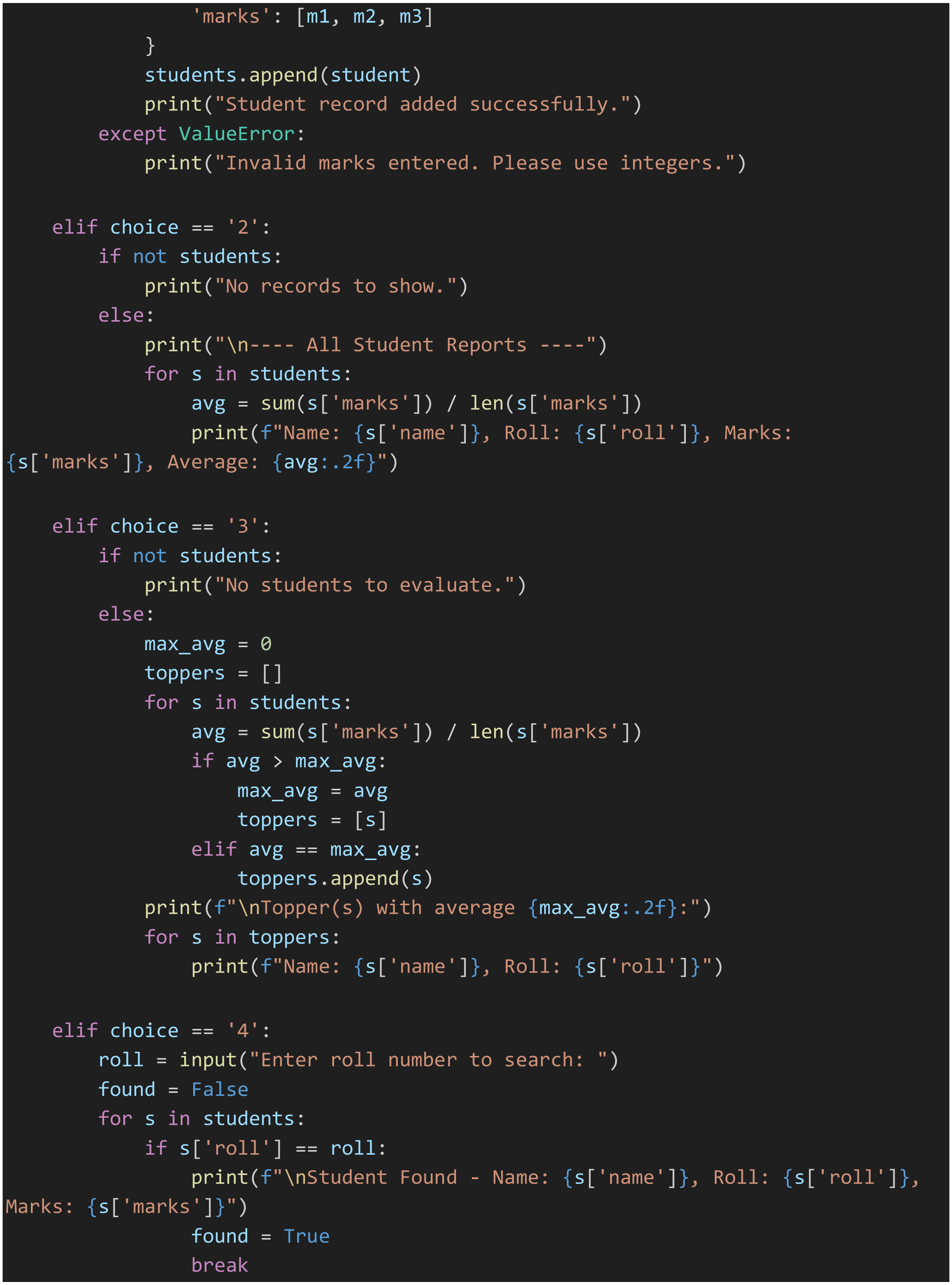
Screenshot:



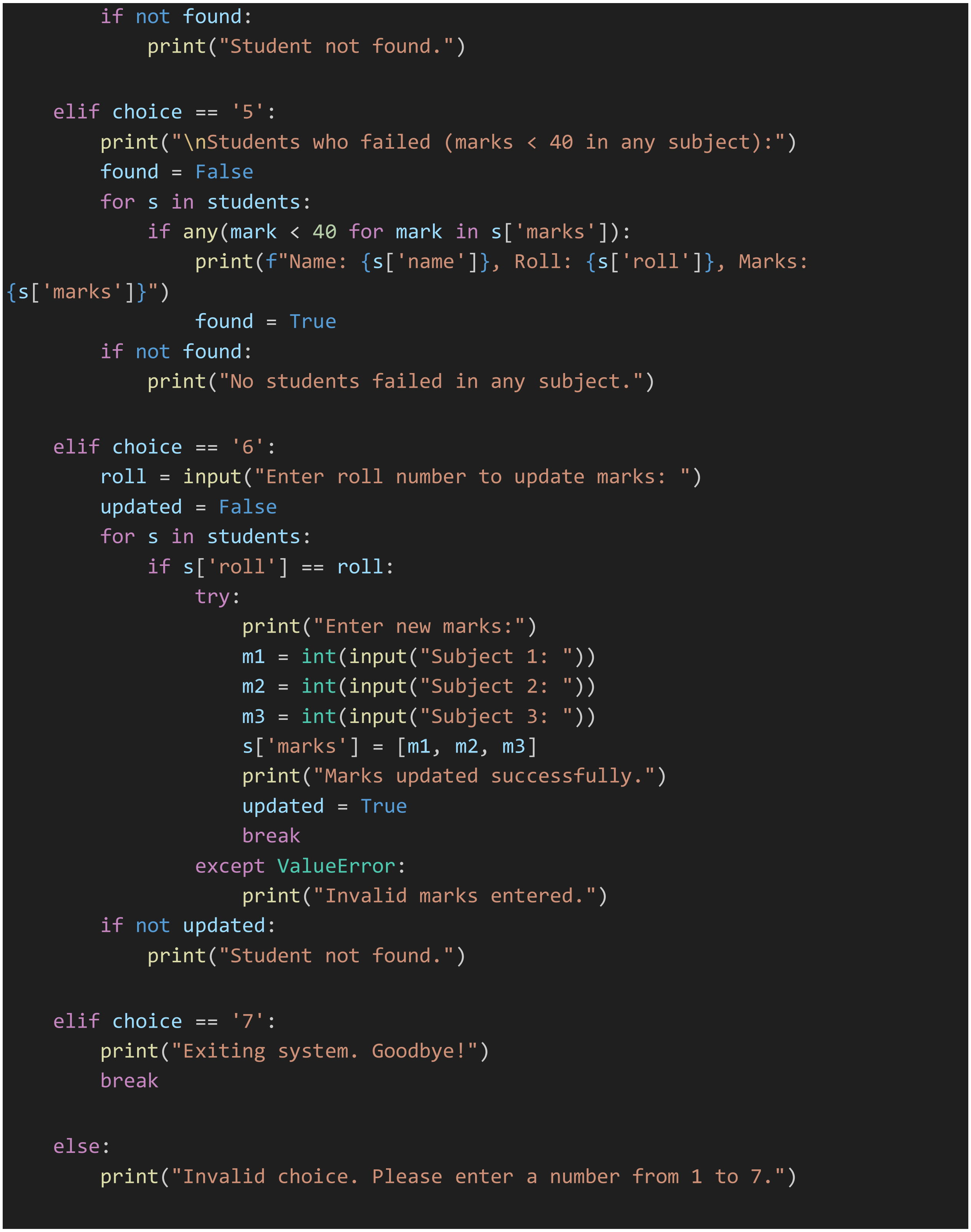
```
Enter your choice (a/b/c/d): 3
Invalid choice.

Choose an option:
a. Add a new product
b. Update price of an existing product
c. Find products within a price range
d. Exit
Enter your choice (a/b/c/d): b
Enter the product name to update: cokasprite
Enter the new price: 200
Updated cokasprite to new price 200.0

Choose an option:
a. Add a new product
b. Update price of an existing product
c. Find products within a price range
d. Exit
Enter your choice (a/b/c/d): d
Exiting program.
PS C:\Users\Pratyush Shrestha\Desktop\Advanced Computer Programming\labwork2>
```

MINI PROJECT: Student Report Card Management System Problem Statement: Design and implement a Student Report Card Management System using Python that allows a teacher to: ● Add new student records (name, roll number, subject-wise marks). ● View the report of all students. ● Display the topper(s) of the class based on average marks. ● Search for a student by roll number. ● Display all students who have failed in one or more subjects. ● Optionally update marks of any student

```python
students = []

while True:
    print("\n===== Student Report Card Management System =====")
    print("1. Add new student record")
    print("2. View report of all students")
    print("3. Display class topper(s)")
    print("4. Search student by roll number")
    print("5. Display students who failed in one or more subjects")
    print("6. Update marks of a student")
    print("7. Exit")
    choice = input("Enter your choice (1-7): ")

    if choice == '1':
        name = input("Enter student name: ")
        roll = input("Enter roll number: ")
        try:
            m1 = int(input("Enter marks in Subject 1: "))
            m2 = int(input("Enter marks in Subject 2: "))
            m3 = int(input("Enter marks in Subject 3: "))
            student = {
                'name': name,
                'roll': roll,
```

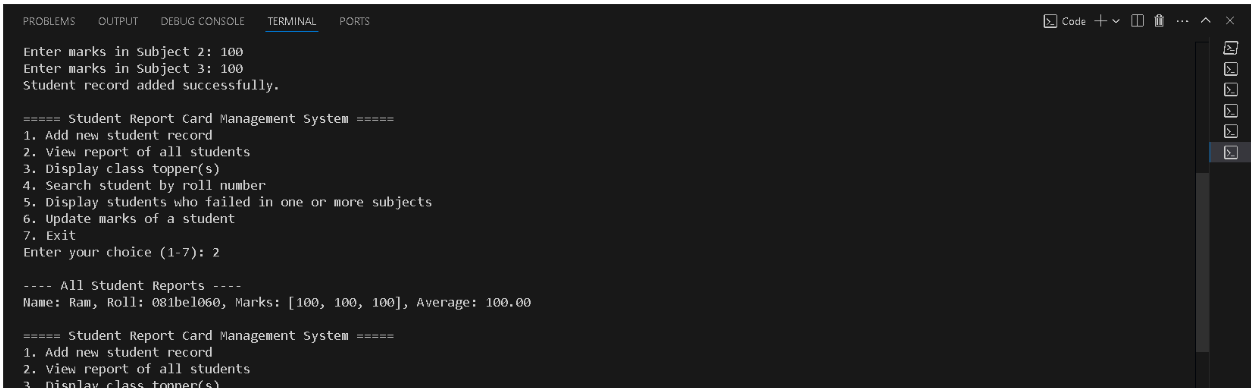


```
                'marks': [m1, m2, m3]
            }
            students.append(student)
            print("Student record added successfully.")
        except ValueError:
            print("Invalid marks entered. Please use integers.")

    elif choice == '2':
        if not students:
            print("No records to show.")
        else:
            print("\n---- All Student Reports ----")
            for s in students:
                avg = sum(s['marks']) / len(s['marks'])
                print(f"Name: {s['name']}, Roll: {s['roll']}, Marks:
{s['marks']}, Average: {avg:.2f}")

    elif choice == '3':
        if not students:
            print("No students to evaluate.")
        else:
            max_avg = 0
            toppers = []
            for s in students:
                avg = sum(s['marks']) / len(s['marks'])
                if avg > max_avg:
                    max_avg = avg
                    toppers = [s]
                elif avg == max_avg:
                    toppers.append(s)
            print(f"\nTopper(s) with average {max_avg:.2f}:")
            for s in toppers:
                print(f"Name: {s['name']}, Roll: {s['roll']}")

    elif choice == '4':
        roll = input("Enter roll number to search: ")
        found = False
        for s in students:
            if s['roll'] == roll:
                print(f"\nStudent Found - Name: {s['name']}, Roll: {s['roll']},
Marks: {s['marks']}")
                found = True
                break
```



```
        if not found:
            print("Student not found.")

    elif choice == '5':
        print("\nStudents who failed (marks < 40 in any subject):")
        found = False
        for s in students:
            if any(mark < 40 for mark in s['marks']):
                print(f"Name: {s['name']}, Roll: {s['roll']}, Marks:
{s['marks']}")
                found = True
        if not found:
            print("No students failed in any subject.")

    elif choice == '6':
        roll = input("Enter roll number to update marks: ")
        updated = False
        for s in students:
            if s['roll'] == roll:
                try:
                    print("Enter new marks:")
                    m1 = int(input("Subject 1: "))
                    m2 = int(input("Subject 2: "))
                    m3 = int(input("Subject 3: "))
                    s['marks'] = [m1, m2, m3]
                    print("Marks updated successfully.")
                    updated = True
                    break
                except ValueError:
                    print("Invalid marks entered.")
        if not updated:
            print("Student not found.")

    elif choice == '7':
        print("Exiting system. Goodbye!")
        break

    else:
        print("Invalid choice. Please enter a number from 1 to 7.")
```

Screenshot:



```
Enter marks in Subject 2: 100
Enter marks in Subject 3: 100
Student record added successfully.

===== Student Report Card Management System =====
1. Add new student record
2. View report of all students
3. Display class topper(s)
4. Search student by roll number
5. Display students who failed in one or more subjects
6. Update marks of a student
7. Exit
Enter your choice (1-7): 2

---- All Student Reports ----
Name: Ram, Roll: 081bel060, Marks: [100, 100, 100], Average: 100.00

===== Student Report Card Management System =====
1. Add new student record
2. View report of all students
3. Display class topper(s)
```