Output:

Enter item price : 25.50
10.00
50.75
Done

Total Bill Amount : ₹86.25
Highest Priced Item : 250.75
Lowest Priced Item : 210.00

DATE:- 13/08/25

Task 4:- Use various data types, list, Tuples and dictionary in python programming

4.a) Shopping Cart Price Calculator (List)

Aim:- To calculate the total bill amount, identify the highest priced item, and find the lowest-priced item from a list of purchased item prices.

Algorithm:-

1. Start the program.
2. Create a list to store item prices.
3. Calculate the sum of all prices in the list to get the total bill amount.
4. Find the maximum value in the list to get the total bill amount.
5. Stop the program.

Program:-

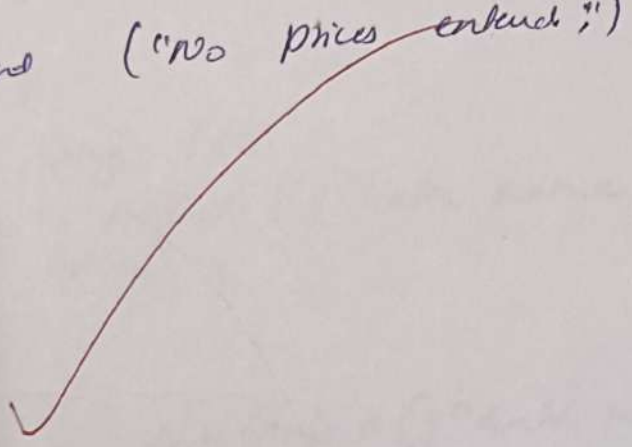
Enter item price (enter 'done' to finish):
15.50
20.00
5.25
done
Total bill amount : 40.75
Highest Priced Item : 20.00
Lowest priced item : 5.25

Program:

```
prices = []
while True:
    try:
        price_input = input("Enter item price")

if prices:
    total_bill = sum(prices)
    highest_price = max(prices)
    lowest_price = min(prices)
    print(f"\n Total Bill Amount : 2)
    print(f"\n Highest Priced Item : 2)
    print f(" Lowest Priced Item : 2)

else:
    print("No prices entered;")
```



Output:-

Enter name for student 1: Rahul

Enter marks for Rahul : 456

Enter name for student 2: Priya

Enter marks for Priya : 380

Student with highest marks: Amit (456 marks)

Students scoring above 400 marks:

- Rahul 456

4b) Student Exam Result (Tuple)

Aim:- To store student names and marks using tuples, then display the student with the highest marks and all students scoring above 400 marks.

Algorithm:

1) Start the program.
2) Iterate through the list of student tuples to find the student with the highest marks.
3) Display the student with the highest marks.
4) Display all student with scoring above 400 marks.
5) End the program.

Program:-

```
students = []
for i in range (5):
    name = input (f"Enter name for student
        {i+1}: ")
while True:
    try:
        marks = int (input (f"Enter marks for {name}: "))
        students. append (name, marks))
        break
    except value error:
        print ("Invalid input. Please enter a number for
            marks: ")
if not found - above - 400:
print ("No students scored above 400 marks:")
```

4.c) Country - capital Pendu

Aim:- To manage country - capital pairs using a python dictionary, allowing addition of new pairs, searching for capitals, and displaying all pairs alphabetically.

Algorithm:

1. Start the program
2. Initialize a dictionary with some pre-defined country - capital pairs.
3. Add new pair : Prompt the user for a new country and its capital, then add it to the dictionary.
4. End the program.

Program:-

```
country_capitals = {"India": "New Delhi", "France":
                    "Paris", "Japan": "Tokyo"}
new_country = input ("Enter new country to add:")
new_capital = input ("Enter capital for {new_country}:")
country_capitals [new_country] = new_capital
print ("Added: {new_country} - {new_capital}")
search_country = input ("Enter country to find
                         its capital:")
```

Output:

Enter new country to add: Germany
Enter capital for germany: Berlin
Added: Germany - Berlin

Enter country to find its capital: France
The capital of France is Paris.

All country - capital Pairs (Alphabetical):

- France: Paris
- Germany: Berlin
- India: New Delhi
- Japan: Tokyo

```
if capital_found:
    print(f"The capital of {search_country} is
      {capital_found}.")
capital_found
else:
    print(f"Capital for {search_country} not found")

print("\n All country - capital Pairs:")
for country, capital in sorted(country_capitals
                               .items()):
    print(f"{country}: {capital}")
```

| VEL TECH - CSE | |
|---|---|
| EX NO. | 5 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 5 |
| RECORD (5) | 5 |
| TOTAL (20) | 25 |
| SIGN WITH DATE | 12/9/25 |

Result: We successfully created a python program for [illegible] and country - capital finder.