Date:- 13/08/2025

Task 4: Shopping cart price calculator

Aim: To find the shopping cart price calculator of the algorithm and program

Algorithm:

1. Start the program
2. Create list to store item prices
3. use the sum() function to calculate the total bill
4. use the max() function to find the highest-priced item
5. use the min() function to find the lowest-priced item
6. Display the shopping cart, total amount, highest, and lowest prices
7. End the program

program:

```
Create a list of prices
shopping_cart = [12.50, 7.99, 24.30, 3.49, 15.20, 9.99]
Calculate total bill
total_bill = sum(shopping_cart)
find highest-priced item
highest_price = max(shopping_cart)
find lowest-priced item
lowest_price = min(shopping_cart)
Display the results
print("Shopping cart prices:", shopping_cart)
print("Total Bill Amount: $", round(total_bill,2))
print("Highest priced item: $", round(higher_price,2))
print("lowest priced item: $", round(lowest_price,2))
```

Result:-

Thus the shopping cart price calculator executed Successfully.

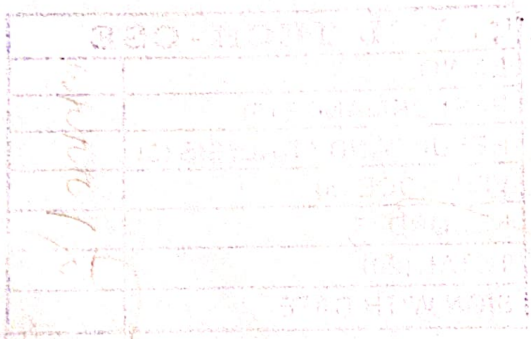
output:

Shopping cart prices: [12.5, 7.99, 24.3, 3.49, 15.2, 9.99]

Total bill Amount: $ 73.47

Highest priced item: $24.3

lowest priced item: $ 3.49.



Task 4.2 Student Example result:-

Aim:- To find the Student Example result with aim, algorithm, program result.

Algorithms:-

1. Start the program
2. create a list of tuples, where each tuple contains
   - Student name
   - Total marks
3. initialize a variable to track the student with the highest marks
4. loop through the list to find
   - The student with the highest marks
   - All students scoring above 400
5. Display the required results
6. End the program

program:

Step 1: Store student data as tuples in a list

```
students = ["Rahul", 456)
            "Anjali", 389),
            "Vikram", 421),
            "Sneha", 478),
            "Aman", 395).
          ]
```

Step 2: Find Student with highest marks.

```
highest_student = students[0]
for student in students:
    if student[1] > highest_student[1]:
    highest_student = student
```

Step 3:

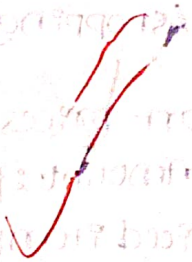
```
above_400_students = []
for student in students:
    if student[1] > 400:
    above_400_students.append(student)
print("Student with highest marks:")
print(f"Name: {highest_student[0]}, marks: {highest_student[1]}
```

Result:

Thus the student Example result program executed Successfully.

Output:

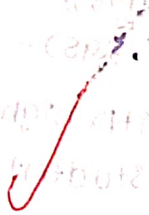
Student with highest marks:
Name: Sneha, marks: 478
Students who scored above 400 marks:
Name: Rahul, marks: 456
Name: Vikram, marks: 421
Name: Sneha, marks: 478.



output:

Enter a new country name: Italy
Enter the capital of Italy : Rome

Enter a country name to search for its capital : France
The capital of France is paris.



Task 4.3: country-capital finder (Dictionary)

Aim:

To write a python program to find country-capital finder.

Algorithm:

1. Start the program
2. initialize a dictionary with some predefined country-capital pairs
3. prompt the user to user to enter a new country and its capital, and add it to the dictionary
4. Ask the user to enter a country name to search for its capital
5. Sort the dictionary by country names and display all pairs.
6. End the program.

Program:

```
initialize dictionary with some country-capital pairs
country_capitals = {"India": "New delhi",
                    "france": "paris"
                    "Japan": "Tokyo"
                    "Germany": "Berlin"
}
new_country = input("Enter a new country name:")
new_capital = input("Enter the capital of {new_country}:")
country_capitals[new_country] = new_capital
print("\nAll country-capital pairs (sorted by country:")
for country in sorted(country_capitals):
  print(f"{country}: {country_capitals[country]}")
```

VEL TECH-CSE	
EX NO.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	25
SIGN WITH DATE	

Result:

Thus country capital-finder program executed successfully.

20/6/25