

Rajalakshmi Engineering College

Name: Ishwaryaa N
Email: 240701196@rajalakshmi.edu.in
Roll no: 240701196
Phone: 9655727550
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: `total_cost(item_cost)`

Input Format

The input consists of a single line containing a positive floating-point number representing the cost of the item.

Output Format

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

Output: Item Cost: \$50.00

Sales Tax Rate: 8.0%

Total Cost: \$54.00

Answer

#

You are using Python

SALES_TAX_RATE=0.08

item_cost=float(input())

def total_cost(item_cost):

 tax_amount=item_cost*SALES_TAX_RATE

 total=item_cost+tax_amount

 return total

total_cost = total_cost(item_cost)

print(f"Item Cost: \${item_cost:.2f}")

print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")

print(f"Total Cost: \${total_cost:.2f}")

Status : Correct

Marks : 10/10

2. Problem Statement

You are tasked with designing a shipping cost calculator program that calculates the shipping cost for packages based on their weight and destination. The program utilizes different shipping rates for domestic, international, and remote destinations. The rates for each destination type are provided as global constants.

Constant Values:

DOMESTIC_RATE = 5.0

INTERNATIONAL_RATE = 10.0

REMOTE_RATE = 15.0

Function Signature: `calculate_shipping(weight, destination)`

Formula: $\text{shipping cost} = \text{weight} * \text{destination rate}$

Input Format

The first line of the input consists of a float representing the weight of the package.

The second line consists of a string representing the destinations (Domestic or International or Remote).

Output Format

The program outputs any one of the following:

1. If the input is valid and the destination is recognized, the output should consist of a single line stating the calculated shipping cost for the given weight and destination in the format: "Shipping cost to [destination] for a [weight] kg package: \$[calculated cost]" with two decimal places.
2. If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."
3. If the input destination is not one of the valid options, print "Invalid destination."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5.5

Domestic

Output: Shipping cost to Domestic for a 5.5 kg package: \$27.50

Answer

#

DOMESTIC_RATE = 5.0

INTERNATIONAL_RATE = 10.0

REMOTE_RATE = 15.0

```
def calculate_shipping(weight, destination):
```

```
    if weight <= 0:
```

```
        print("Invalid weight. Weight must be greater than 0.")
```

```
        return None
```

```
    if destination == "Domestic":
```

```
        rate = DOMESTIC_RATE
```

```
    elif destination == "International":
```

```
        rate = INTERNATIONAL_RATE
```

```
    elif destination == "Remote":
```

```
        rate = REMOTE_RATE
```

```
    else:
```

```
        print("Invalid destination.")
```

```
        return None
```

```
    return weight * rate
```

```
weight_input = input()
```

```
destination = input()
```

```
try:
```

```
    weight = float(weight_input)
```

```
    shipping_cost = calculate_shipping(weight, destination)
```

```
except ValueError:
```

```
    print("Invalid weight. Weight must be greater than 0.")
```

```
if shipping_cost is not None:
```

```
    print(f"Shipping cost to {destination} for a {weight} kg package:
```

`${shipping_cost:.2f}")`

Status : Correct

Marks : 10/10

3. Problem Statement

Meena is analyzing a list of integers and needs to count how many numbers in the list are even and how many are odd. She decides to use lambda functions to filter the even and odd numbers from the list.

Write a program that takes a list of integers, counts the number of even and odd numbers using lambda functions, and prints the results.

Input Format

The first line contains an integer n , representing the number of integers in the list.

The second line contains n space-separated integers.

Output Format

The first line of output prints an integer representing the count of even numbers.

The second line of output prints an integer representing the count of odd numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
12 34 56 78 98 65 23

Output: 5
2

Answer

```
# Function to count even and odd numbers
def count_even_odd(n, nums):
```

```
even_count = len(list(filter(lambda x: x % 2 == 0, nums)))
odd_count = len(list(filter(lambda x: x % 2 != 0, nums)))

print(even_count)
print(odd_count)
n = int(input())
nums = list(map(int, input().split()))
```

```
count_even_odd(n, nums)
```

Status : Correct

Marks : 10/10

4. Problem Statement

Implement a program for a retail store that needs to find the highest even price in a list of product prices. Your goal is to efficiently determine the maximum even price from a series of product prices. Utilize the `max()` inbuilt function in the program.

For example, if the prices are 10 15 24 8 37 16, the even prices are 10 24 8 16. So, the maximum even price is 24.

Input Format

The input consists of a series of product prices separated by a space.

The prices should be entered as a space-separated string of numbers.

Output Format

If there are even prices in the input, the output prints "The maximum even price is: " followed by the maximum even price.

If there are no even prices in the input, the output prints "No even prices were found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10 15 24 8 37 16

Output: The maximum even price is: 24

Answer

You are using Python

```
prices_input = input()
```

```
prices = list(map(int, prices_input.split()))
```

```
even_prices = [price for price in prices if price % 2 == 0]
```

```
if even_prices:
```

```
    max_even = max(even_prices)
```

```
    print(f"The maximum even price is: {max_even}")
```

```
else:
```

```
    print("No even prices were found")
```

Status : Correct

Marks : 10/10