

Rajalakshmi Engineering College

Name: Divya darshini S
Email: 241501051@rajalakshmi.edu.in
Roll no: 241501051
Phone: 6383045036
Branch: REC
Department: I AIML FA
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results



NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 31.5

Section 1 : Coding

1. Problem Statement

Arjun is working on a mathematical tool to manipulate lists of numbers. He needs a program that reads a list of integers and generates two lists: one containing the squares of the input numbers, and another containing the cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the input list using lambda functions.

Input Format

The input consists of a single line of space-separated integers representing the list of input numbers.

Output Format

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3

Output: [1, 4, 9]

[1, 8, 27]

Answer

```
input_numbers_str = input()
```

```
numbers = [int(num) for num in input_numbers_str.split()]
```

```
square = lambda x: x**2
```

```
cube = lambda x: x**3
```

```
squared_numbers = list(map(square, numbers))
```

```
cubed_numbers = list(map(cube, numbers))
```

```
print(squared_numbers)
```

```
print(cubed_numbers)
```

Status : Correct

Marks : 10/10

2. 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

```
def find_highest_even_price(prices_str):  
  
    prices = [int(price) for price in prices_str.split()]  
    even_prices = [price for price in prices if price % 2 == 0]  
  
    if even_prices:  
        return f"The maximum even price is: {max(even_prices)}"  
    else:  
        return "No even prices were found"  
  
input_prices = input()  
  
result = find_highest_even_price(input_prices)  
print(result)
```

Status : Correct

Marks : 10/10

3. Problem Statement

Develop a text analysis tool that needs to count the occurrences of a specific substring within a given text string.

Write a function `count_substrings(text, substring)` that takes two inputs: the text string and the substring to be counted. The function should count how many times the substring appears in the text string and return the count.

Function Signature: `count_substrings(text, substring)`

Input Format

The first line of the input consists of a string representing the text.

The second line consists of a string representing the substring.

Output Format

The output should display a single line of output containing the count of occurrences of the substring in the text string.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: programming is fun and programming is cool
programming

Output: The substring 'programming' appears 2 times in the text.

Answer

```
text=input()
substring=input()
count=text.count(substring)
print(f"The substring '{substring}' appears {count} times in the text.")
```

Status : Correct

Marks : 10/10

4. 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

#

```
def calculate_shipping(weight,destination):
    if weight <= 0:
        print("Invalid weight. Weight must be positive.")
        return None

    destination = destination.strip().lower()

    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 = float(input())
destination = input()

shipping_cost = calculate_shipping(weight,destination)

if shipping_cost is not None:
    print(f"Shipping cost to {destination} for a {weight} kg package:
    ${shipping_cost:.2f}")
```

Status : Partially correct

Marks : 1.5/10