

Rajalakshmi Engineering College

Name: Saii Anish R
Email: 241501175@rajalakshmi.edu.in
Roll no: 241501175
Phone: 8438920387
Branch: REC
Department: I AI & ML FB
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 : 40

Section 1 : Coding

1. Problem Statement

Create a program for a mathematics competition where participants need to find the smallest positive divisor of a given integer n . Your program should efficiently determine this divisor using the `min()` function and display the result.

Input Format

The input consists of a single positive integer n , representing the number for which the smallest positive divisor needs to be found.

Output Format

The output prints the smallest positive divisor of the input integer in the format:
"The smallest positive divisor of $[n]$ is: [smallest divisor]"

Refer to the sample output for the exact format.

Sample Test Case

Input: 24

Output: The smallest positive divisor of 24 is: 2

Answer

```
def find_smallest_divisor(n):
    if n == 1:
        return 1
    divisors = []
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            divisors.append(i)
    return min(divisors) if divisors else n

def main():
    n = int(input())
    result = find_smallest_divisor(n)
    print(f"The smallest positive divisor of {n} is: {result}")

if __name__ == "__main__":
    main()
```

Status : Correct

Marks : 10/10

2. 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

```
nums = list(map(int, input().split()))  
print(list(map(lambda x: x**2, nums)), list(map(lambda x: x**3, nums)))
```

Status : Correct

Marks : 10/10

3. 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
```

```
try:  
    weight = float(input())  
    destination = 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

4. 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

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

Status : Correct

Marks : 10/10