Rajalakshmi Engineering College

Name: Kanishka S
Email: 240701227@rajalakshmi.edu.in
Roll no: 2116240701227
Phone: 8825651385
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

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 smallest_divisor(n):
    divisors = []
    for i in range(2, n + 1):
        if n % i == 0:
            divisors.append(i)
    if len(divisors)>0:
        smallest = min(divisors)
    else:
        smallest = n
    print(f"The smallest positive divisor of {n} is: {smallest}")


n = int(input())
smallest_divisor(n)
```

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: shipping cost = weight * 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":
        return weight * DOMESTIC_RATE
    elif destination == "International":
        return weight * INTERNATIONAL_RATE
    elif destination == "Remote":
        return weight * REMOTE_RATE
    else:
        print("Invalid destination.")
        return None
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 : Correct **Marks : 10/10**

3. Problem Statement

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z)Uppercase letters (A-Z)Digits (0-9)Special characters (from string.punctuation, e.g. @, !, #, $)

Input Format

The input consists of a single string representing the user's password.

Output Format

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length ≥ 6 and at least 2 different character types, the output prints "<password> is Moderate"

If Password length ≥ 10 and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: password123

Output: password123 is Moderate

Answer

```
import string

password = input()
length = len(password)
lowercase = any(c.islower() for c in password)
uppercase = any(c.isupper() for c in password)
digit = any(c.isdigit() for c in password)
special = any(c in string.punctuation for c in password)

types = sum([lowercase, uppercase, digit, special])

if length>=10 and types==4:
    print(password + " is Strong")
elif length >= 6 and types >= 2:
    print(password + " is Moderate")
else:
    print(password + " is Weak")
```

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

```
prices_str = input()
prices = [int(x) for x in
prices_str.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**