Rajalakshmi Engineering College

Name: RANNESH KHUMAR B R
Email: 240701422@rajalakshmi.edu.in
Roll no: 2116240701
Phone: 9042350670
Branch: REC
Department: I CSE FD
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

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

```
#

# You are using Pythonshippinng_cost=None
dr=5.0
ir=10.0
rr=15.0
shipping_cost=None
weight=float(input())
destination=input()
```

```
if weight<=0:
    print("Invalid weight. Weight must be greater than 0.")
elif destination=='Domestic':
        shipping_cost=weight*dr

elif destination=='International':
     shipping_cost=weight*ir

elif destination=='Remote':
      shipping_cost=weight*rr
else:
   print("Invalid 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***

2.   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

```python
# You are using Python
s=input()
lc=0
uc=0
dc=0
sc=0
length=len(s)
for i in range(len(s)):
    if(s[i].isupper()):
        uc+=1
    elif(s[i].islower()):
        lc+=1
    elif(s[i].isdigit()):
        dc+=1
    else:
        sc+=1
types=0
if lc>0:
    types+=1
if uc>0:
    types+=1
```

```
if dc>0:
   types+=1
if dc>0:
   types+=1

if(length<6 or types<2):
   print(s,"is weak")
elif(length>=10 and types==4):
   print(s,"is strong")
else:
   print(s,"is moderate")
```

Status : Correct   **Marks : 10/10**

3. 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

#

```python
# You are using Python
item_cost=float(input())
SALES_TAX_RATE=0.08
total_cost=item_cost+(item_cost*SALES_TAX_RATE)
def total_cost(item_cost):
    total_cost=item_cost+(item_cost*SALES_TAX_RATE)
    return total_cost


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*

4. 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.

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

```python
# You are using Python
l=[]
n=int(input())
for i in range(2,n+1):
    if(n%i==0):
        l.append(i)

print(f"The smallest positive divisor of {n} is:{min(l)}")
```

Status : Correct   *Marks : 10/10*