WS3 – LOOPS

*NAME: PREM DHARSHAN D ROLL NO. :22PW29*

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 15:42:56 2023

@author: 22pw29

"""

"""1. Write a function that takes the lengths of the two shorter sides of a right triangle as its parameters. Return the hypotenuse of the triangle, computed using Pythagorean theorem, as the function’s result. Include a main program that reads the lengths of the shorter sides of a right triangle from the user, uses your function to compute the length of the hypotenuse, and displays the result."""

from math import sqrt

def calcHypotenuse(s1, s2):

return sqrt(s1\*s1 + s2\*s2)

def main():

s1 = int(input("Enter the length opposite of the triangle (units): "))

s2 = int(input("Enter the length adjacent side of the triangle (units): "))

s3 = calcHypotenuse(s1, s2)

print(f"\nThe length of the hypotenuse is {s3} units.\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 15:56:55 2023

@author: 22pw29

"""

"""2. Write a function that takes two positive integers that represent the numerator and denominator of a fraction as its only two parameters. The body of the function should reduce the fraction to lowest terms and then return both the numerator and denominator of the reduced fraction as its result. For example, if the parameters passed to the function are 6 and 63 then the function should return 2 and 21. Include a main program that allows the user to enter a numerator and denominator. Then your program should display the reduced fraction."""

from math import gcd

def reducedFraction(nume: int, deno: int) -> (int, int):

return (nume//gcd(nume, deno), deno//gcd(nume, deno))

def main() -> None:

n = int(input("Enter a numerator :"))

d = int(input("Enter a denominator :"))

print(f"The reduced fraction is {reducedFraction(n, d)[0]}/{reducedFraction(n, d)[1]}")

return None

if \_\_name\_\_=="\_\_main\_\_":

main()# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 16:00:56 2023

@author: 22pw29

"""

"""3. There are three seating categories at a stadium. For a softball game, Class A seats cost $15, Class B seats cost $12, and Class C seats cost $9. Write a program that asks how many tickets for each class of seats were sold, and then displays the amount of income generated from ticket sales."""

def ticketsSold() -> tuple:

classA = int(input("Enter the number of tickets sold in Class A: "))

classB = int(input("Enter the number of tickets sold in Class B: "))

classC = int(input("Enter the number of tickets sold in Class C: "))

return (classA, classB, classC)

def incomeCalculater() -> None:

A, B, C = ticketsSold()

print(f"\nThe amount of income generated from ticket sales is ${A\*15 + B\*12 + C\*9}.")

return

def main() -> None:

incomeCalculater()

return

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 16:14:39 2023

@author: 22pw29

"""

"""4. A painting company has determined that for every 115 square feet of wall space, one gallon of paint and eight hours of labor will be required. The company charges $20.00 per hour for labor. Write a program that asks the user to enter the square feet of wall space to be painted and the price of the paint per gallon. The program should display the following data:

a. • The number of gallons of paint required

b. • The hours of labor required

c. • The cost of the paint

d. • The labor charges

e. • The total cost of the paint job

"""

from math import ceil

def userData():

area = int(input("Enter the square feet of wall space to be painted: "))

price = float(input("Enter the price of the paint per gallon: $"))

return (area, price)

def display():

area, price = userData()

print(f"\nThe number of gallons of paint required is {ceil(area/115)}.")

print(f"The hours of labor required is {(area/115)\*8:.1f}.")

print(f"The cost of the paint is ${ceil(area/115)\*price:.2f}.")

print(f"The labor charges area ${(area/115)\*8\*20:.2f}.")

print(f"The total cost of the paint job is ${ceil(area/115)\*price+(area/115)\*8\*20:.2f}")

return

def main():

display()

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 16:33:09 2023

@author: 22pw29

"""

"""5. A prime number is an integer greater than 1 that is only divisible by one and itself. Write a function that determines whether or not its parameter is prime, returning True if it is, and False otherwise. Write a main program that reads an integer from the user and displays a message indicating whether or not it is prime"""

def isPrime(num:int) -> bool:

if (num <= 1):

return False

for i in range(2, int(num\*\*0.5)+1):

if (num%i == 0):

return False

return True

def main():

if (isPrime(int(input("Enter a number: ")))):

print("The entered number is a Prime Number")

else:

print("The entered number is not a Prime Number")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 1 17:01:38 2023

@author: 22pw29

"""

"""6. In a particular jurisdiction,

taxi fares consist of a base fare of $4.00, plus $0.25 for every 140 meters travelled.

Write a function that takes the distance travelled (in kilometers) as its only parameter and returns the total fare as its only result.

Write a main program that demonstrates the function"""

def fareCalc(distance) -> float:

return 4 + (distance\*1000/140) \* 0.25

def main() -> None:

distance = float(input("Enter the distance travelled: "))

print(f"The total fare for the distance travelled is {fareCalc(distance):.2f}.")

if \_\_name\_\_ == "\_\_main\_\_" :

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 15 11:36:45 2023

@author: 22pw29

"""

"""7. In a particular jurisdiction, taxi fares consist of a base fare of $4.00, plus $0.25 for every 140 meters travelled. Write a function that takes the distance travelled (in kilometers) as its only parameter and returns the total fare as its only result. Write a main program that demonstrates the function."""

def taxiFare(distance: int) -> float:

return 4 + ((distance//140)\*0.25)

def main() -> None:

dist: int = int(input("Enter distance in kilometers : "))

print(f"Taxi fare for {dist} km is ${taxiFare(dist):.2f}.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 15 11:43:56 2023

@author: 22pw29

"""

"""8. Write a function that takes three numbers as parameters, and returns the median value of those parameters as its result. Include a main program that reads three values from the user and displays their median."""

def median(a: int, b: int, c: int) ->int:

lst =[a,b,c]

lst.remove(max([a,b,c]))

lst.remove(min([a,b,c]))

return lst[0]

def main() -> None:

a:int = int(input("Enter first number: "))

b:int = int(input("Enter second number: "))

c:int = int(input("Enter third number: "))

print(f"Median is {median(a,b,c)}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 15 11:50:12 2023

@author: 22pw29

"""

"""9. If you have 3 straws, possibly of differing lengths, it may or may not be possible to lay them down so that they form a triangle when their ends are touching. For example, if all of the straws have a length of 6 inches. then one can easily construct an equilateral triangle using them. However, if one straw is 6 inches. long, while the other two are each only 2 inches. long, then a triangle cannot be formed. In general, if any one length is greater than or equal to the sum of the other two then the lengths cannot be used to form a triangle. Otherwise they can form a triangle. Write a function that determines whether or not three lengths can form a triangle. The function will take 3 parameters and return a Boolean result. In addition, write a program that reads 3 lengths from the user and demonstrates the behaviour of this function."""

from typing import List

def validTriangle(sides: List[int]) -> bool:

if (sides[0] > sides[1]+sides[2]): return False

elif (sides[1] > sides[2]+sides[0]): return False

elif (sides[2] > sides[0]+sides[1]): return False

else: return True

def main():

side1: int = int(input("Enter side 1: "))

side2: int = int(input("Enter side 2: "))

side3: int = int(input("Enter side 3: "))

if validTriangle([side1, side2, side3]):

print("The 3 sides form a triangle")

else:

print("The 3 sides do not form a triangle")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 15 11:58:28 2023

@author: 22pw29

"""

"""10. Write a function named precedence that returns an integer representing the precedence of a mathematical operator. A string containing the operator will be passed to the function as its only parameter. Your function should return 1 for + and -, 2 for \* and /, and 3 for ˆ. If the string passed to the function is not one of these operators then the function should return -1. Include a main program that reads an operator from the user and either displays the operator’s precedence or an error message indicating that the input was not an operator. """

def precedence(operator: str) -> int:

if operator in '+-': return 1

elif operator in '\*/': return 2

elif operator == '^': return 3

else: return -1

def main() -> None:

a: str = input("Enter the operator: ")

if (precedence(a) == -1):

raise TypeError ("Input is not an operator")

else:

print(f"Precedence of {a} is {precedence(a)}.")

if \_\_name\_\_=="\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 15 12:01:26 2023

@author: 22pw29

"""

"""11. Words like first, second and third are referred to as ordinal numbers. In this exercise, you will write a function that takes an integer as its only parameter and returns a string containing the appropriate English ordinal number as its only result. Your function must handle the integers between 1 and 12 (inclusive). It should return an empty string if a value outside of this range is provided as a parameter. Include a main program that demonstrates your """

def ordinalNumber(num: int) -> str:

if num == 1:

return "First"

elif num == 2:

return "Second"

elif num == 3:

return "Third"

elif num == 4:

return "Fourth"

elif num == 5:

return "Fifth"

elif num == 6:

return "Sixth"

elif num == 7:

return "Seventh"

elif num == 8:

return "Eighth"

elif num == 9:

return "Nineth"

elif num == 10:

return "Tenth"

elif num == 11:

return "Eleventh"

elif num == 12:

return "Twelth"

else:

return ""

def main() -> None:

import os

os.system("pip install prettytable")

from prettytable import PrettyTable

table = PrettyTable()

table.field\_names = ["Month Number", "Ordinal Number"]

for i in range(1,13):

table.add\_row([i, ordinalNumber(i)])

print(table)

return None

if \_\_name\_\_ == '\_\_main\_\_':

main()

# -\*- coding: utf-8 -\*-

"""

Created on Mon Feb 6 09:49:59 2023

@author: 22pw29

"""

"""

12. write a function that determines whether or not a password is good. We will define a good password to be a one that is at least 8 characters long and contains at least one uppercase letter, at least one lowercase letter, and at least one number. Your function should return true if the password passed to it as its only parameter is good. Otherwise it should return false. Include a main program that reads a password from the user and reports whether or not it is good.

"""

import os

os.system("pip install re")

from re import search

def isValidPassword(password):

while (len(password) >= 8):

if (search("[a-z]", password) and

search("[A-z]", password) and

search("[0-9]", password)):

return True

else:

return False

else:

return False

def main():

password = input("\nEnter a password to check validity: ")

if (isValidPassword(password)):

print("\nThe entered password is a Valid Password")

else:

print("\nThe entered password is not a Valid Password")

return

if \_\_name\_\_ == "\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Mon Feb 13 09:49:51 2023

@author: 22pw29

"""

"""13. Write a function that determines how many days there are in a particular month. Your function will take two parameters: The month as an integer between 1 and 12, and the year as a four digit integer. Ensure that your function reports the correct number of days in February for leap years. Include a main program that reads a month and year from the user and displays the number of days in that month."""

from calendar import month\_name

def isLeap(year: int) -> bool:

if (year % 4 == 0):

if (year % 100 == 0):

if (year % 400 == 0):

return True

else:

return False

else:

return True

else:

return False

def daysInAMonth(month: int, year: int) -> int:

if month in [1,3,5,7,8,10,12]:

return 31

elif month in [4,6,9,11]:

return 30

else:

if isLeap(year):

return 29

else:

return 28

def main() -> None:

month: int = int(input("Enter the month number (1-12): "))

year: int = int(input('Enter the year: '))

print (f"{month\_name[month]} month in {year} has {daysInAMonth(month, year)} days")

if \_\_name\_\_ == '\_\_main\_\_':

main()

# -\*- coding: utf-8 -\*-

"""

Created on Mon Feb 13 08:43:51 2023

@author: 22pw29

"""

"""14. Many recipe books still use cups, tablespoons and teaspoons to describe the volumes of ingredients used when cooking or baking. While such recipes are easy enough to follow if you have the appropriate measuring cups and spoons, they can be difficult to double, triple or quadruple when cooking Christmas dinner for the entire extended family. For example, a recipe that calls for 4 tablespoons of an ingredient requires 16 tablespoons when quadrupled. However, 16 tablespoons would be better expressed (and easier to measure) as 1 cup. Write a function that expresses an imperial volume using the largest units possible. The function will take the number of units as its first parameter, and the unit of measure (cup, tablespoon or teaspoon) as its second parameter. Return a string representing the measure using the largest possible units as the function’s only result.For example, if the function is provided with parameters representing 59 teaspoons then it should return the string “1 cup, 3 tablespoons, 2 teaspoons”. Hint: One cup is equivalent to 16 tablespoons. One tablespoon is equivalent to 3 teaspoons."""

def unitConverter(teaspoon):

cup = teaspoon//48

tablespoon = (teaspoon%48)//3

teaspoon = (teaspoon%48)%3

return cup, tablespoon, teaspoon

def main():

teaspoons = int(input("Enter the number of teaspoons: "))

units = unitConverter(teaspoons)

print(f"{teaspoons} teaspoons is equivalent to {units[0]} Cups, {units[1]} Tablespoons and {units[2]} Teaspoons.\n")

if \_\_name\_\_=="\_\_main\_\_":

main()

# -\*- coding: utf-8 -\*-

"""

Created on Mon Feb 13 08:59:04 2023

@author: 22pw29

"""

"""15. Write a recursive function that implements the recurrence relation to calculate the nth power of an integer value x. Both n and x should be parameters to the function. The return value of the function will be value of xn. For example: if n=4 and x=2 the function should return 24=16. """

def expo(x: int, n: int) -> int:

if (n != 0):

return x \* expo(x, n-1)

else :

return 1

def main():

x = int(input("Enter the x: "))

n = int(input("Enter the n: "))

print("\nThe value of x^n is", expo(x, n), ".\n")

return None

if \_\_name\_\_ == "\_\_main\_\_":

main()