1. **C**) %
2. **B**) 0
3. **C**) 24
4. **A**) 2
5. **D**) 6
6. **C**) the finally block will be executed no matter if the try block raises an error or not
7. **A**) It is used to raise an exception
8. **C**) in defining a generator
9. **A**) \_abc (starts with an underscore and followed by letters) C) abc2 (starts with a letter and followed by letters and a digit)
10. **A**) yield and B) raise

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Welcome!")

print("This program calculates the factorial of any number! ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n-1)

# Take input from the user

while True:

num\_str = input("Enter a number (positive integer): ").strip()

try:

# Convert the input to float

num = int(num\_str)

# Check if the number is a positive integer

if num >= 0:

result = factorial(num)

print("The factorial of", num, "is", result)

break

else:

print("Factorial is not defined for negative numbers.")

except ValueError:

print("Invalid input. Please enter a positive integer.")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Welcome!")

print("This program finds whether a number is prime or composite! ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def is\_prime(num):

"""

Check if a number is prime or composite.

Returns True if the number is prime, False otherwise.

"""

if num <= 1:

return False

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

if num % i == 0:

return False

return True

# Prompt the user to enter a positive integer

num\_str = input("Enter a positive integer: ")

try:

# Convert the user input to an integer

num = int(num\_str)

# Check if the number is prime or composite using the is\_prime() function

if is\_prime(num):

print(num, "is a prime number.")

else:

print(num, "is a composite number.")

except ValueError:

# Display an error message if the input is not a valid positive integer

print("Invalid input. Please enter a positive integer.")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Welcome!")

print("This program finds whether a string is a palindrome or not! ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def is\_palindrome(string):

"""

Check if a string is a palindrome.

Returns True if the string is a palindrome, False otherwise.

"""

# Remove any whitespace and convert the string to lowercase

string = string.replace(" ", "").lower()

# Reverse the string

reversed\_string = string[::-1]

# Compare the original string with the reversed string

if string == reversed\_string:

return True

else:

return False

# Prompt the user to enter a string

string = input("Enter a string: ")

# Check if the string is a palindrome using the is\_palindrome() function

if is\_palindrome(string):

print("The string is a palindrome.")

else:

print("The string is not a palindrome.")

# This program gets the third side of a right-angled triangle from two given sides.

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Welcome!")

print("Get the third side of a right angle triangle given lenth of 2 sides! ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

import math

# This function calculates the length of the third side of a right-angled triangle.

def calculate\_third\_side(a, b):

"""

Calculate the length of the third side of a right-angled triangle.

Returns the length of the third side.

"""

# Use the Pythagorean theorem: c^2 = a^2 + b^2

c = math.sqrt(a\*\*2 + b\*\*2)

return c

# Prompt the user to enter the lengths of the two sides

a = float(input("Enter the length of the first side: "))

b = float(input("Enter the length of the second side: "))

# Calculate the length of the third side using the calculate\_third\_side() function

c = calculate\_third\_side(a, b)

# Print the length of the third side

print("The length of the third side is:", c)

def count\_characters(string):

"""

Count the frequency of each character in a given string.

Prints the character and its frequency.

"""

# Create an empty dictionary to store the character frequencies

char\_freq = {}

# Iterate over each character in the string

for char in string:

# Check if the character is already in the dictionary

if char in char\_freq:

# Increment the frequency count of the character

char\_freq[char] += 1

else:

# Add the character to the dictionary with a frequency count of 1

char\_freq[char] = 1

# Print the character and its frequency

for char, freq in char\_freq.items():

print(f"Character '{char}' appears {freq} times")

# Prompt the user to enter a string

input\_string = input("Enter a string: ")

# Call the count\_characters() function to count the character frequencies

count\_characters(input\_string)

