In [3]:
```python
# 11.Write a python program to find the factorial of a number
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)

# Input: Get a number from the user
num = int(input("Enter a number to find its factorial: "))

# Calculate and display the factorial
result = factorial(num)
print(f"The factorial of {num} is: {result}")
```

```
Enter a number to find its factorial: 4
The factorial of 4 is: 24
```

In [4]:
```python
# 12.Write a python program to find whether a number is prime or composite.

def is_prime(number):
    if number <= 1:
        return False
    elif number == 2:
        return True
    elif number % 2 == 0:
        return False
    else:
        # Check for factors up to the square root of the number
        for i in range(3, int(number**0.5) + 1, 2):
            if number % i == 0:
                return False
        return True

# Input: Get a number from the user
num = int(input("Enter a number: "))

# Check and display whether the number is prime or composite
if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is a composite number.")
```

```
Enter a number: 7
7 is a prime number.
```

In [5]:
```python
# 14.Write a Python program to get the third side of right-angled triangle from two
import math

def find_third_side(side1, side2):
    # Calculate the hypotenuse (third side) using the Pythagorean theorem
    third_side = math.sqrt(side1**2 + side2**2)
    return third_side

# Input: Get two sides of the right-angled triangle from the user
side1 = float(input("Enter the length of the first side: "))
side2 = float(input("Enter the length of the second side: "))

# Calculate and display the third side
third_side = find_third_side(side1, side2)
print(f"The length of the third side is: {third_side}")
```

```
Enter the length of the first side: 9
Enter the length of the second side: 16
The length of the third side is: 18.35755975068582
```

In [2]:
```python
# 15.Write a python program to print the frequency of each of the characters presen
def character_frequency(string):
    # Create an empty dictionary to store character frequencies
    frequency_dict = {}

    # Count the frequency of each character in the string
    for char in string:
        if char in frequency_dict:
            frequency_dict[char] += 1
        else:
            frequency_dict[char] = 1

    # Display the character frequencies
    for char, frequency in frequency_dict.items():
        print(f"Character '{char}' appears {frequency} times.")

# Input: Get a string from the user
input_string = input("Enter a string: ")

# Print the frequency of each character in the string
character_frequency(input_string)
```

```
Enter a string: madhu
Character 'm' appears 1 times.
Character 'a' appears 1 times.
Character 'd' appears 1 times.
Character 'h' appears 1 times.
Character 'u' appears 1 times.
```

In [6]:
```python
# 13.Write a python program to check whether a given string is palindrome or not.
def is_palindrome(s):
    # Convert the string to lowercase and remove spaces
    s = s.lower().replace(" ", "")

    # Compare the original string with its reverse
    return s == s[::-1]

# Input: Get a string from the user
string = input("Enter a string: ")

# Check and display whether the string is a palindrome or not
if is_palindrome(string):
    print(f"{string} is a palindrome.")
else:
    print(f"{string} is not a palindrome.")
```

```
Enter a string: radar
radar is a palindrome.
```

In [ ]: