1. C) %
2. B) 0
3. C) 24
4. D) 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 and C) abc2
10. A) yield and B) raise
11.

In [2]:
```python
def factorial(n):
    """
    This function computes the factorial of a given number using recursion.
    """
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Taking user input
num = int(input("Enter a number: "))

# Checking if input is positive or negative
if num < 0:
    print("Factorial cannot be found for negative numbers.")
elif num == 0:
    print("Factorial of 0 is 1.")
else:
    print("Factorial of", num, "is", factorial(num))
```

```
Enter a number: 4
Factorial of 4 is 24
```

12.

```
In [3]: def is_prime(n):
            """
            This function checks whether a given number is prime or composite.
            """
            if n <= 1:
                return False
            elif n <= 3:
                return True
            elif n % 2 == 0 or n % 3 == 0:
                return False
            i = 5
            while i * i <= n:
                if n % i == 0 or n % (i+2) == 0:
                    return False
                i += 6
            return True

        # Taking user input
        num = int(input("Enter a number: "))

        # Checking if input is prime or composite
        if is_prime(num):
            print(num, "is a prime number.")
        else:
            print(num, "is a composite number.")
```

```
Enter a number: 4
4 is a composite number.
```

13.

```
In [4]: def isPalindrome(s):
            return s == s[::-1]

        s = "malayalam"
        ans = isPalindrome(s)

        if ans:
            print("Yes")
        else:
            print("No")
```

```
Yes
```

14.

In [11]:
```python
test_str = "kingsslayer"

all_freq = {}

for i in test_str:
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1

# printing result
print("Count of all characters  :\n "+ str(all_freq))
```

```
Count of all characters  :
 {'k': 1, 'i': 1, 'n': 1, 'g': 1, 's': 2, 'l': 1, 'a': 1, 'y': 1, 'e': 1, 'r': 1}
```

15.

In [5]:
```python
from math import sqrt
print("Input lengths of shorter triangle sides:")
a = float(input("a: "))
b = float(input("b: "))
c = sqrt(a**2 + b**2)
print("The length of the hypotenuse is:", c )
```

```
Input lengths of shorter triangle sides:
a: 33
b: 33
The length of the hypotenuse is: 46.66904755831214
```