

ANS1. C) %

ANS2. B) 0

ANS3. C) 24

ANS4. A) 2

ANS5. D) 6

ANS6. C) the finally block will be executed no matter if the try block raises an error or not.

ANS7. A) It is used to raise an exception.

ANS8. C) in defining a generator

ANS9. A) \_abc & C) abc2

ANS10. A) yield & B) raise

ANS11.

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n-1)
```

```
num = 7
```

```
print(f"The factorial of {num} is {factorial(num)}")
```

ANS12.

```
def is_prime(n):
```

```
    if n < 2:
```

```
        return False
```

```
    for i in range(2, int(n ** 0.5) + 1):
```

```
        if n % i == 0:
```

```
            return False
```

```
    return True if n != 1 else False
```

```
num = 7
```

```
if is_prime(num):
```

```
    print(f"{num} is a prime number")
```

```
else:
```

```
    print(f"{num} is a composite number")
```

ANS13.

```
def is_palindrome(s):  
    return s == s[::-1]  
string = "racecar"  
if is_palindrome(string):  
    print(f"{string} is a palindrome")  
else:  
    print(f"{string} is not a palindrome")
```

ANS 14.

```
def pythagoras(opposite_side, adjacent_side):  
    hypotenuse = (opposite_side ** 2 + adjacent_side ** 2) ** 0.5  
    return hypotenuse  
opposite_side = 3  
adjacent_side = 4  
print(f"The length of the third side of the right-angled triangle with opposite side {opposite_side} and  
adjacent side {adjacent_side} is {pythagoras(opposite_side, adjacent_side)}")
```

ANS15.

```
def char_frequency(string):  
    frequency = {}  
    for char in string:  
        if char in frequency:  
            frequency[char] += 1  
        else:  
            frequency[char] = 1  
    return frequency  
string = "example string"  
print(f"The frequency of each character in '{string}' is:")  
for char, count in char_frequency(string).items():  
    print(f"{char}: {count}")
```