1. Which of the following operators is used to calculate remainder in a division?

A) # B) & C) % D)

In [ ]:
```
C) %
```

2. In python 2//3 is equal to?

A) 0.666 B) 0 C) 1 D) 0.67

In [ ]:
```
A) 0.666
```

3. In python, 6<<2 is equal to?

A) 36 B) 10 C) 24 D) 45

In [ ]:
```
C) 24
```

4. In python, 6&2 will give which of the following as output?

A) 2 B) True C) False D) 0

In [ ]:
```
 D) 0
```

5. In python, 6|2 will give which of the following as output?

A) 2 B) 4 C) 0 D) 6

In [ ]:
```
B) 4
```

6. What does the finally keyword denotes in python?

A) It is used to mark the end of the code B) It encloses the lines of code which will be executed if any error occurs while executing the lines of code in the try block. C) the finally block will be executed no matter if the try block raises an error or not. D) None of the above

In [ ]:
```
 C) the finally block will be executed no matter if the try block raises an error or not.
```

7. What does raise keyword is used for in python?

A) It is used to raise an exception. B) It is used to define lambda function C) it's not a keyword in python. D) None of the above

In [ ]:
```
A) It is used to raise an exception.
```

8. Which of the following is a common use case of yield keyword in python?

A) in defining an iterator B) while defining a lambda function C) in defining a generator D) in for loop

In [ ]:
```
A) in defining an iterator
```

Q9 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

9. Which of the following are the valid variable names? A) _abc B) 1abc C) abc2 D) None of the above

In [ ]:
```
 A) _abc
```

10. Which of the following are the keywords in python?

A) yield B) raise C) look-in D) all of the above

In [ ]:
```
D) all of the above
```

11. Write a python program to find the factorial of a number

In [*]:
```
def factorial_recursive(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial_recursive(n - 1)

num = int(input("Enter a number: "))
if num < 0:
    print("Factorial is not defined for negative numbers.")
else:
    result = factorial_recursive(num)
    print(f"The factorial of {num} is {result}")
```

```
Enter a number:
```

12. Write a python program to find whether a number is prime or composite.

In [*]:
```
def is_prime(number):
    if number <= 1:
        return False
    if number <= 3:
        return True
    if number % 2 == 0 or number % 3 == 0:
        return False
    i = 5
    while i * i <= number:
        if number % i == 0 or number % (i + 2) == 0:
            return False
        i += 6
    return True

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

if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is a composite number.")
```

13. Write a python program to check whether a given string is palindrome or not

In [*]:
```
def is_palindrome(s):
    s = s.lower()  # Convert the string to lowercase to make the comparison case-insensit
    s = ''.join(filter(str.isalnum, s))  # Remove non-alphanumeric characters
    return s == s[::-1]  # Compare the string with its reverse

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

if is_palindrome(input_string):
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

14. Write a Python program to get the third side of right-angled triangle from two given sides

In [*]:
```
import math

def calculate_third_side(side1, side2):
    third_side = math.sqrt(side1**2 + side2**2)
    return third_side

def main():
    try:
        side1 = float(input("Enter the length of the first side: "))
        side2 = float(input("Enter the length of the second side: "))

        if side1 <= 0 or side2 <= 0:
            print("Side lengths must be positive numbers.")
        else:
            third_side = calculate_third_side(side1, side2)
            print("The length of the third side is:", third_side)

    except ValueError:
        print("Invalid input. Please enter valid numerical values.")

if __name__ == "__main__":
    main()
```

15. Write a python program to print the frequency of each of the characters present in a given string

In [*]:
```
def character_frequency(input_string):
    frequency = {}

    for char in input_string:
        if char in frequency:
            frequency[char] += 1
        else:
            frequency[char] = 1

    return frequency

def main():
    input_string = input("Enter a string: ")
    frequency = character_frequency(input_string)

    print("Character frequencies:")
    for char, count in frequency.items():
        print(f"'{char}': {count}")

if __name__ == "__main__":
    main()
```