Pattern Printing

1. **Print the following patterns:**

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

2. **Print the following number patterns:**

```
1
2 3
4 5 6
7 8 9 10
```

3. **Print the following triangle number pattern:**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4. **Print the following increasing digit patterns:**

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
```

5. **Print the following alphabet patterns:**

```
A
A B
A B C
A B C D
A B C D E
```

6. **Print alphabet patterns in ascending order:**

```
A
B C
D E F
G H I J
K L M N O
```

7. **Print inverted number patterns:**

   1 2 3 4 5

   1 2 3 4

   1 2 3

   1 2

   1

~~**8. Print pyramid pattern:**~~

   ~~**   *~~**

   ~~** * * ***~~

   ~~*** * * * ***~~

Turtle Shape Patterns

9. **Draw Pentagon using Python Turtle.**
10. **Draw Hexagon using Python Turtle.**
11. **Draw Octagon using Python Turtle.**
12. **Draw Nonagon using Python Turtle.**
13. **Draw Decagon using Python Turtle.**
14. **Generate any pattern of your choice using Python Turtle.**

Number and List Operations

15. **Check whether a given number is prime or not.**
16. **Create a Fibonacci series of `n` terms.**
17. **Create 5 different lists using list comprehension.**

18. **Create a list, add 25 random numbers, and print the largest number.**
19. **Check whether a given number is present in the list or not.**
20. **Demonstrate 10 methods of a list.**
21. **Add two lists (element-wise addition).**
22. **Implement a stack using a list.**
23. **Sort a list of numbers in ascending order using functions.**

String Operations

24. **Find duplicate characters in a string.**
25. **Check whether a given string is a palindrome.**
26. **Capitalize the first character in a given sentence (using built-in methods).**
27. **Convert a string to a list.**
28. **Sort alphabetically the words in a sentence (use `split()`, `sort()`, and `join()` methods).**
29. **Demonstrate 10 functions of the string class.**
30. **Count the number of vowels in a given string.**
31. **Check if two strings are anagrams.**
32. **Demonstrate string slicing.**
33. **Check if a substring is present in a string.**

Dictionary Operations

34. Create a dictionary with multiple key-value pairs.
35. Demonstrate 10 methods of dictionary.
36. Create 3 dictionaries using dictionary comprehension.

File Handling

37. Read the contents of a file.
38. Write into a file.
39. Count occurrences of a word in a text file.
40. Count total blank spaces in a file and replace them with hashtags.
41. Copy content from one file to another.
42. Copy an image file.
43. Convert lowercase characters from one file to uppercase in another.

~~44. Write into a file, read the content, and append until the user enters `@`.~~

45. Count the number of lines, words, and characters in a text file.

~~46. Convert uppercase characters from one file to lowercase in another.~~

Tuple Operations

47. Create a tuple with numbers till 20, print half in one line and the rest in the next.

48. Create a tuple with numbers from 1 to 100 and another tuple with only even numbers.

49. Concatenate two tuples.

50. Accept elements as a tuple and display their sum and average.

51. Find the first occurrence of an element in a tuple.

52. Pickle list, dictionary, tuple, and string.

53. Demonstrate 2 methods of the tuple class.

Set Operations

54. Demonstrate 5 methods of a set.

Socket Programming

55. Python Server sending details to the client.

56. Two-way communication (Server-client both sending messages).

~~57. Chat server in Python (until the client sends `over`).~~

~~58. Send a file from client to server.~~

~~59. Implement a secure server in Python.~~

Encryption and Games

60. Implement Caesar Cipher using Python.

61. Implement Caesar Cipher using a dictionary.

62. Create a Rock-Paper-Scissors game using a list with a loop until the player says `n`.

~~**63. Implement Snake and Ladder game using python dictionary**~~

Database Operations (SQLite3)

64. Demonstrate the following:

- **a. Create a table.**
- **b. Insert operation.**
- **c. Select operation.**
- **d. Update operation.**
- **e. Delete table.**

~~Random Programs~~

~~**65. Pizza Order Program**~~

~~**Write a program in Python to order pizza, specifying size (large, medium, small), ask the user if they want to add toppings (example: pepperoni), and also if the user wants extra cheese. (Don't forget to calculate and print the bill)**~~

~~**66. Split Bill with 12% Tip**~~

~~**Write a program in Python to calculate split bill with a 12% tip.**~~

~~67. Days, Weeks, and Months Left Until 90~~

~~Write a program in Python to calculate the number of months, weeks, and days left before you turn 90.~~

~~68. FizzBuzz Program~~

~~Write a FizzBuzz program in Python (print “Fizz for multiples of 3, “Buzz” for multiples of 5, and “FizzBuzz” for multiples of both)~~

~~69. Write a program that converts temperature from Celsius to Fahrenheit and vice versa.~~

~~70. Write a program that calculates BMI based on the user’s weight (in kg) and height (in meters).~~

~~71. Write a program that converts a given number of seconds into hours, minutes, and seconds.~~

~~72. Write a program that asks for the user's birth year and calculates how old they are (considering the current date).~~

~~73. Write a program that calculates the area of a rectangle, circle, and triangle, based on user input.~~

~~74. Write a Python program that demonstrates the any five functions of the os module~~

~~75. Write a program that calculates the area of a circle if you are given the circumference.~~

~~76. Write a program that takes a string (a sentence or a phrase) and returns the acronym. For example, the input "National Aeronautics and Space Administration" should output "NASA".~~

~~77. Create a program that calculates the total earnings based on hourly wage and hours worked. For example, if someone earns 150/hour~~

~~and works 40 hours, the program will output the total earnings for the week.~~

~~78. Write a program that checks if a password meets certain criteria:~~

  ~~a. At least 8 characters long~~

  ~~b. Contains at least one number~~

  ~~c. Contains at least one special character (@, #, etc.)~~

Object-Oriented Programming (OOP)

79. Demonstrate OOP concepts in Python:

  a. Class & Object.

  b. Constructor.

  c. Class variable and Class method.

80. Demonstrate OOP concepts:

  a. Inheritance.

  b. Super method.

  c. Polymorphism.

~~81. Demonstrate OOP concepts:~~

  ~~a. Constructor with inheritance.~~

  ~~b. Method overloading.~~

  ~~c. Method overriding.~~

82. Implement a simple banking application using OOP.

~~83. Demonstrate Python packages and modules:~~

  ~~a. Create 3+ classes, put them in one package, and import them into another program.~~

Data Visualization

84. Visualize the sales trend over last 6 months of a company by creating a bar graph. Plot the months on the X-axis and plot the Sales on the Y-axis.

85. Plot the following types of graphs:

    a. Bar Graph

    b. Histogram

    c. Scatter Plot

    d. Pie Chart

NumPy Operations

86. Create 1D, 2D, and 3D arrays using NumPy.

87. Demonstrate the following NumPy functions:

    a. \`dtype\`

    b. \`reshape\`

    c. \`arange\`

    d. \`empty\`

    e. \`ones\`

    f. \`zeros\`

    g. \`asarray\`

88. Print the indices of the max element in an array and the max element of each row/column.

89. Find the minimum, maximum, and sum of a NumPy array.

90. **Find the square root and standard deviation of an array using NumPy.**

91. **Sort an array:**

    a. **Entire array**

    b. **Row-wise (`axis=0`)**

    c. **Column-wise (`axis=1`)**

92. **Find the mean of an array in a given list.**

93. **Add rows to a NumPy array.**

94. **Add columns to a NumPy array.**

95. **Reverse an array using NumPy.**

96. **Multiply two matrices using NumPy.**

97. **Add two matrices using NumPy.**

98. **Subtract two matrices using NumPy.**

99. **Transpose a matrix using NumPy.**

~~Pandas~~

~~**100. Simulating the Law of Large Numbers Using Pandas and Matplotlib for Random Sampling, Mean Calculation, and Visualization.**~~