For loop

1. Write a program to print numbers from 1 to 10 using a for loop.
2. Print the even numbers between 1 and 20 using a for loop.
3. Print the multiplication table of 5 using a for loop.
4. Find the sum of the first 10 natural numbers using a for loop.
5. Print the square of numbers from 1 to 10 using a for loop.
6. Write a program to reverse a given string using a for loop.
7. Count the number of vowels in a given string using a for loop.
8. Print the factorial of a given number using a for loop.
9. Print Fibonacci series up to **n** terms using a for loop.
10. Find the largest number in an array using a for loop.
11. Write a program to check if a given number is a prime number using a for loop.
12. Print all prime numbers between 1 and 50 using a for loop.
13. Write a program to count occurrences of a specific character in a string using a for loop.
14. Write a program to print a right-angled triangle pattern using a for loop:

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

15. Create a number pyramid using a for loop:

```
1
121
12321
1234321
```

1. Print all odd numbers between 1 and 50 using a for loop.
2. Write a program to print numbers from **n** to 1 using a for loop.
3. Print the first **n** multiples of a given number.
4. Calculate the sum of all even numbers between 1 and 100.
5. Find the product of all numbers from 1 to **n** using a for loop.
6. Reverse a number using a for loop (e.g., input: 1234, output: 4321).

7. Count the number of digits in a given number using a for loop.
8. Print the sum of digits of a number using a for loop.
9. Check if a number is an Armstrong number using a for loop (e.g., $153 = 1^3 + 5^3 + 3^3$).
10. Write a program to print the **GCD (Greatest Common Divisor)** of two numbers using a for loop.
11. Print all perfect numbers between 1 and 1000 using a for loop (A number is perfect if the sum of its factors, excluding itself, is equal to the number).
12. Write a program to check if a number is palindrome using a for loop (e.g., 121 is a palindrome).
13. Write a program to generate a **Floyd's Triangle** using a for loop:

    1

    2 3

    4 5 6

    7 8 9 10

14. Generate the following pattern using a for loop:

    1

    2 3

    4 5 6

    7 8 9 10

1. Print all multiples of 3 from 1 to 100 using a for loop.
2. Write a program to print numbers in descending order from **n** to 1.
3. Print the ASCII values of characters from 'A' to 'Z' using a for loop.
4. Print the sum of the first **n** odd numbers.
5. Find the sum of all numbers divisible by 5 between 1 and 100.
6. Print all prime factors of a given number using a for loop.
7. Write a program to count the number of words in a string using a for loop.
8. Find the **LCM (Least Common Multiple)** of two numbers using a for loop.
9. Print the reverse of an array using a for loop.
11. Find the sum of digits of all numbers from 1 to **n** using a for loop.
12. Convert a binary number to decimal using a for loop.
13. Generate the Fibonacci sequence up to **n** terms using a for loop.
14. Print all Armstrong numbers between 1 and 1000 using a for loop.

15. Find the smallest and largest number in a given list using a for loop.
16. Write a program to print a **Hollow Square Pattern** using a for loop:

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

17. Generate a **Diamond Pattern** using a for loop:

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

1. Print the following pattern using nested for loops:

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

2. Print a right-angled triangle pattern:

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

3. Print a square pattern with numbers:

```
1111
2222
3333
4444
```

4. Print a decreasing number triangle:

```
55555
4444
```

333

22

1

5. Print a multiplication table for numbers 1 to 5 using nested for loops.
6. Print a pyramid pattern:

*

***

*****

****

7. Print an inverted pyramid pattern:

****

*****

***

*

8. Print a Floyd's triangle:

1

2 3

4 5 6

7 8 9 10

9. Print a hollow square pattern:

*****

* *

* *

*****

10. Print a number pyramid:

1

121

12321

1234321

11. Print a diamond pattern using nested for loops:

*

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

12. Print a checkerboard pattern using * and spaces:

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

13. Print an **hourglass pattern** using nested for loops:

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

14. Print an **X-pattern** using nested for loops:

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

15. Print a **hollow diamond pattern** using nested loops:

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

```
* *
*
```

16. Print the **chessboard pattern** using # and spaces:

```
# # # #
# # # #
# # # #
# # # #
```

1. Print the following square pattern:

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

2. Print a **right-angled triangle pattern**:

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

3. Print a **reverse right-angled triangle pattern**:

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

4. Print a **hollow square pattern**:

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

5. Print a **number pattern**:

```
1
12
123
```

1234

6. Print a **pyramid pattern**:

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

7. Print a **hollow pyramid pattern**:

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

8. Print a **diamond pattern**:

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

9. Print a **number pyramid**:

```
   1
  121
 12321
1234321
```

10. Print **Floyd's triangle**:

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

11. Print **Pascal's Triangle** up to **n** rows.
12. Print an **hourglass pattern**:

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

13. Print an **X-pattern**:

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

14. Print a **hollow diamond pattern**:

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

15. Generate a **checkerboard pattern** using * and spaces:

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

16. Print **Zig-Zag Pattern**:

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

17. Find and print all **prime numbers between 1 and 100** using nested loops.
18. Print a **Chessboard Pattern** using # and spaces:

```
# # # #
# # # #
# # # #
# # # #
```

19. Print the **Hollow Butterfly Pattern**:

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

20. Print an **inverted hourglass pattern**:

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