

1. Write a Python program to generate and print all prime numbers up to 100.

```
Prime numbers up to 100:
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

2. Write a Python program to find and print all Armstrong numbers in the range of 1 to 1000.

```
Armstrong numbers from 1 to 1000:
```

```
1 2 3 4 5 6 7 8 9 153 370 371 407
```

3. Write a Python program to generate and print the Fibonacci series up to a given number n.

```
Enter the number of terms in the Fibonacci series: 10
```

```
0 1 1 2 3 5 8 13 21 34
```

4. Write a Python program to calculate the factorial of a given number without using recursion.

```
Factorial of 5 is: 120
```

5. Write a Python program to generate a random two-digit number and allow the user to guess it. Provide feedback on whether the guess is too high, too low, or correct.

```
I've selected a random two-digit number between 10 and 99.
```

```
Can you guess what it is?
```

```
Enter your guess: 45
```

```
Too low.
```

```
Enter your guess: 65
```

```
Too high.
```

```
Enter your guess: 55
```

```
Too high.
```

```
Enter your guess: 50
```

```
Too low.
```

```
Enter your guess: 52
```

```
Congratulations! You guessed it: 52
```

6. Write a Python program that prints numbers from 1 to 100. For multiples of 3, print "Fizz" instead of the number. For multiples of 5, print "Buzz" instead of the number. For numbers that are multiples of both 3 and 5, print "FizzBuzz."

```
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19
```

7. Write a Python program to solve a quadratic equation of the form  $ax^2 + bx + c = 0$ , where the user inputs the values of a, b, and c, and the program calculates the roots using the quadratic formula.

For example, if the user enters `a = 1`, `b = -3`, and `c = 2`, the output will be:

```
Real roots: Root 1 = 2.0 Root 2 = 1.0
```

8. Write a Python program to calculate the future value of an investment based on the principal amount, interest rate, and time period.

For example, if the user enters `principal = 1000`, `rate_of_interest = 0.05` (5%), and `time_period = 3`, the output will be:

```
The future value of the investment is: 1157.625
```

9. Write a Python program that generates and displays a multiplication table up to 10x10. The program should create a neatly formatted table with row and column labels and lines separating rows and columns. Use f-strings for string formatting to ensure a clear and organized presentation of the table. The table should look like this:

Multiplication Table:										
	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Your program should achieve this output by efficiently using loops and f-strings for formatting.

10. Write a Python program that prints a pyramid pattern using asterisks (\*). The program should take an integer 'n' as input, where 'n' represents the number of rows in the pyramid. Your task is to print the pattern based on the input value 'n', where each row should have a pyramid-like structure with ascending and descending asterisks.

For example, if the user enters '5' as the value of 'n', the program should print the following pattern:

