1. Write a NumPy program to append values to the end of an array

```
import numpy as np
arr = np.array([1,2,3,4,5])
print(f"Array before append: {arr}")
arr = np.append(arr, [6,7,8,9])
print(f"Array after append: {arr}")
```

```
Array before append: [1 2 3 4 5]
Array after append: [1 2 3 4 5 6 7 8 9]
```

2. Write a program to print the sum of first N natural numbers by using while loop.

```
num = int(input("Enter a number: "))
i = 1
sum = 0
while i <= num:
  sum += i
  i +=1
print(f"The sum of all {num} number is {sum}")
```

```
Enter a number: 5
The sum of all 5 number is 15
```

3. Write a program to print the sum of first n natural numbers by using recursive function.

```
num = int(input("Enter a number: "))
def sum_of_n(n):
  if n == 1:
    return 1
  else:
    return n+sum_of_n(n-1)
print(f"The sum of all {num} number is {sum_of_n(num)}")
```

```
Enter a number: 5
The sum of all 5 number is 15
```

4. Write a NumPy program to convert Centigrade degrees into Fahrenheit degrees. Centigrade values are stored in a NumPy array.

```
import numpy as np
centigrade = np.array([0.0, 12.0, 45.12, 65.21, 100.0])
fahrenheit = (centigrade * 9/5) + 32
print(f"Centigrade: {centigrade}")
print(f"Fahrenheit: {fahrenheit}")
```

```
Centigrade: [  0.    12.    45.12  65.21 100.  ]
Fahrenheit: [ 32.     53.6   113.216 149.378 212.   ]
```

5. Write a program to print the cube of first N natural numbers in a single line.

```
print([x**3 for x in range(1, int(input("Enter a number: "))+1)])
```

```
Enter a number: 5
[1, 8, 27, 64, 125]
```

6. Write a NumPy program to find common values between two arrays.

```
import numpy as np


array1 = np.array([1, 2, 3, 4, 5])
array2 = np.array([4, 5, 6, 7, 8])

# Find common values
common_values = np.intersect1d(array1, array2)
```

```
print("Common values:", common_values)
```

```
Common values: [4 5]
```

7. Write a program to accept a number between 1 to 12 and display corresponding month and days in that month of a calendar.

```
# Program to display month and days based on input number

months = [
    ("January", 31),
    ("February", 28),  # Not considering leap years here
    ("March", 31),
    ("April", 30),
    ("May", 31),
    ("June", 30),
    ("July", 31),
    ("August", 31),
    ("September", 30),
    ("October", 31),
    ("November", 30),
    ("December", 31)
]

# Accept user input
num = int(input("Enter a number between 1 and 12: "))

# Check and display result
if 1 <= num <= 12:
    month, days = months[num - 1]
    print(f"Month: {month}, Days: {days}")
else:
    print("Invalid input. Please enter a number between 1 and 12.")
```

```
Enter a number between 1 and 12: 5
Month: May, Days: 31
```

8. Write a NumPy program to find the indices of the maximum and minimum values along the given axis of an array.

```
import numpy as np

# Sample 2D array
array = np.array([[10, 50, 30],
                  [60, 20, 90],
                  [40, 80, 70]])

# Find indices of maximum values along axis 0 (column-wise)
max_indices_axis0 = np.argmax(array, axis=0)
# Find indices of minimum values along axis 0 (column-wise)
min_indices_axis0 = np.argmin(array, axis=0)

# Find indices of maximum values along axis 1 (row-wise)
max_indices_axis1 = np.argmax(array, axis=1)
# Find indices of minimum values along axis 1 (row-wise)
min_indices_axis1 = np.argmin(array, axis=1)

print("Original array:")
print(array)

print("\nMax indices along axis 0 (columns):", max_indices_axis0)
print("Min indices along axis 0 (columns):", min_indices_axis0)

print("\nMax indices along axis 1 (rows):", max_indices_axis1)
print("Min indices along axis 1 (rows):", min_indices_axis1)
```

```
Original array:
[[10 50 30]
 [60 20 90]
 [40 80 70]]

Max indices along axis 0 (columns): [1 2 1]
Min indices along axis 0 (columns): [0 1 0]
```

```
Max indices along axis 1 (rows): [1 2 1]
Min indices along axis 1 (rows): [0 1 0]
```

9. Write a program to accept a number between 1 to 7 and display corresponding day in a week.

```
num = int(input("Enter a number between 1 to 7: "))
day = ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
for x in range(1,8):
  if x == num:
    print(f"Day: {day[x-1]}")
  else:
    print("Invalid input. Please enter a number between 1 and 7.")
    break
```

```
Enter a number between 1 to 7: 80
Invalid input. Please enter a number between 1 and 7.
```

10. Write a NumPy program to get the sum of values of the elements that are bigger than 10 in a given array.

```
import numpy as np
arr = np.array([5,8,10,78,20,22,2,3])
max_10 = np.array([])
for x in arr:
  if x > 10:
    max_10 = np.append(max_10,x)
print("Array element greater than 10: ",max_10)
print(f"Sum of all greater than 10 element: {np.sum(max_10)}")
```

```
Array element greater than 10:  [78. 20. 22.]
Sum of all greater than 10 element: 120.0
```

```
import numpy as np
arr = np.array([5,8,10,78,20,22,2,3])
max_10 = np.array([])
max_10 = arr[arr>10]
print("Array element greater than 10: ",max_10)
print(f"Sum of all greater than 10 element: {sum(max_10)}")
```

```
Array element greater than 10:  [78 20 22]
Sum of all greater than 10 element: 120
```

11. Write a program to create a user defined list of 5 elements, then after input another number and check this no. is present in that list or not.

```
lst = []
i = 0
while i < 5:
  num = int(input(f"Enter the list element at index_{i+1}: "))
  lst.append(num)
  i += 1
print("The list: ",lst)
#check whether the new number is in list
while True:
    num = int(input("\nEnter a number to check (or -1 to exit): "))
    if num == -1:
        print("Exiting check...")
        break
    if num in lst:
        print(f"The number {num} is in the list.")
    else:
        print("Item not found.")
```

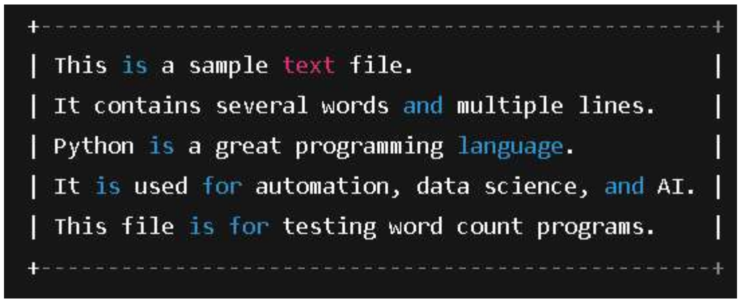
```
Enter the list element at index_1: 8
Enter the list element at index_2: 20
Enter the list element at index_3: 36
Enter the list element at index_4: 25
Enter the list element at index_5: 36
The list:  [8, 20, 36, 25, 36]

Enter a number to check (or -1 to exit): 2
Item not found.
```

```
Enter a number to check (or -1 to exit): 36
The number 36 is in the list.

Enter a number to check (or -1 to exit): -1
Exiting check...
```

12. Write a Python program that takes a text file as input and returns the number of words of a given text file



```
+-----------------------------------------------------------+
| This is a sample text file.                               |
| It contains several words and multiple lines.             |
| Python is a great programming language.                   |
| It is used for automation, data science, and AI.          |
| This file is for testing word count programs.             |
+-----------------------------------------------------------+
```

file.txt

```
path = input("Enter the file path: ")
try:
    with open(path,'r') as file:
        text = file.read()
        words = text.split()
        print(f"Number of words {len(words)}")
except FileNotFoundError as e:
    print("File not found",e)
```

Output:

Enter the file path: ~file.txt

Number of word 36

13. Write a program to take the percentage of a student and prints its grade according to below criteria:

-(Grade A -> greater than 90, Grade B -> greater than 80, Grade C -> greater than 70, Grade D -> greater than 60, Grade E -> greater than 50, Grade F -> less than 50).

```
# Take percentage input from the user
percentage = float(input("Enter the student's percentage: "))

# Determine the grade based on the given criteria
if percentage > 90:
    grade = 'A'
elif percentage > 80:
    grade = 'B'
elif percentage > 70:
    grade = 'C'
elif percentage > 60:
    grade = 'D'
elif percentage > 50:
    grade = 'E'
else:
    grade = 'F'

# Output the result
print(f"The student's grade is: {grade}")
```

```
Enter the student's percentage: 85
The student's grade is: B
```

14. Write a Python program to read a file line by line store it into an array.

```
# Ask user for the file path
file_path = input("Enter the file path: ")

try:
    with open(file_path, 'r') as file:
        lines = file.readlines()  # Read all lines into a list

    # Remove newline characters from each line (optional)
    #lines = [line.strip() for line in lines]

    print("Contents of the file as an array:")
    print(lines)

except FileNotFoundError:
    print("Error: File not found.")
```

Output:

Enter the file path: ~file.txt

Contents of the file as an array: ['This is a sample text file.\n', 'It contains several words and multiple lines.\n', 'Python is a great programming language.\n', 'It is used for automation, data science, and AI.\n', 'This file is for testing word count programs.\n']

15. Write a program to check whether a number is Armstrong or not.

```
# Take input from the user
num = int(input("Enter a number: "))
original_num = num

# Find the number of digits
num_digits = len(str(num))

# Check if the number is Armstrong
result = 0
while num > 0:
    digit = num % 10
    result += digit ** num_digits
    num //= 10

# Print result
if result == original_num:
    print(f"{original_num} is an Armstrong number.")
else:
    print(f"{original_num} is not an Armstrong number.")
```

```
Enter a number: 153
153 is an Armstrong number.
```

16. Write a program to write data into a file and print its content on console.

```
# Data to be written to the file
data = """Hello, this is a sample text.
This is a Python program to write data to a file.
And this is another line of text."""

# Writing data to a file
with open("sample_file.txt", "w") as file:
    file.write(data)

# Reading the content of the file and printing it
with open("sample_file.txt", "r") as file:
    content = file.read()
    print("Content of the file:")
    print(content)
```

```
Content of the file:
Hello, this is a sample text.
This is a Python program to write data to a file.
And this is another line of text.
```

17. Write a program to print the factorial of a number by using recursive function.

```
# Recursive function to find factorial
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

# Take input from user
num = int(input("Enter a number to find its factorial: "))

# Check for valid input
if num < 0:
    print("Factorial is not defined for negative numbers.")
else:
    result = factorial(num)
    print(f"The factorial of {num} is: {result}")
```

```
Enter a number to find its factorial: 5
The factorial of 5 is: 120
```

18. Write a python program to count the number of even and odd numbers from a series of numbers given by user.

```
# Take input from the user
numbers = input("Enter a series of numbers separated by spaces: ")

# Split the input string into a list of integers
num_list = list(map(int, numbers.split()))

# Initialize counters
even_count = 0
odd_count = 0

# Count even and odd numbers
for num in num_list:
    if num % 2 == 0:
        even_count += 1
    else:
        odd_count += 1

# Print the results
print(f"Total even numbers: {even_count}")
print(f"Total odd numbers: {odd_count}")
```

```
Enter a series of numbers separated by spaces: 1 2 3 4 5 6 7 9
Total even numbers: 3
Total odd numbers: 5
```

19. Write a program to check whether a number is prime or not.

```
# Take input from user
num = int(input("Enter a number to check if it's prime: "))

# Prime number check
if num <= 1:
    print(f"{num} is not a prime number.")
else:
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            print(f"{num} is not a prime number.")
            break
    else:
        print(f"{num} is a prime number.")
```

```
Enter a number to check if it's prime: 29
29 is a prime number.
```

20. Write a program to check whether a number is palindrome or not.

```
num = int(input("Enter a number to check if it's a palindrome: "))

# Convert the number to a string
num_str = str(num)

# Check if the number is the same forwards and backwards
if num_str == num_str[::-1]:
    print(f"{num} is a palindrome number.")
else:
    print(f"{num} is not a palindrome number.")
```

```
Enter a number to check if it's a palindrome: 12321
12321 is a palindrome number.
```

21. Write a program to print the prime numbers between a range given by user.

```
  # Take input from user for the range
start = int(input("Enter the start of the range: "))
end = int(input("Enter the end of the range: "))

# Loop through each number in the range
for num in range(start, end + 1):
    # Prime number check
    if num <= 1:
        continue  # Skip numbers less than or equal to 1
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            break  # Not prime, break inner loop
    else:
        # If no divisor found, num is prime
        print(f"{num} is a prime number.")
```

```
Enter the start of the range: 10
Enter the end of the range: 30
11 is a prime number.
13 is a prime number.
17 is a prime number.
19 is a prime number.
23 is a prime number.
29 is a prime number.
```

22. Write a program to create a list of first N even natural numbers where N is given by user

```
lst = []
for x in range(1, int(input("Enter an number: "))+1):
  lst.append(x)
print(f"The list {lst}")
```

```
Enter an number: 5
The list [1, 2, 3, 4, 5]
```

23. Write a program to print first 10th Fibonacci series.

```
# Initialize the first two Fibonacci numbers
a, b = 0, 1

# Print the first 10 Fibonacci numbers
print("First 10 Fibonacci numbers:")
for _ in range(10):
    print(a, end=" ")
    # Update the values of a and b
    a, b = b, a + b
```

```
First 10 Fibonacci numbers:
0 1 1 2 3 5 8 13 21 34
```

24. Write a program to compute simple interest and compound interest.

Formulas:

- **Simple Interest (SI)**:

$$\text{SI} = \frac{P \times R \times T}{100}$$

  Where:

  - P = Principal amount
  - R = Rate of interest per annum
  - T = Time in years

- **Compound Interest (CI)**:

$$\text{CI} = P\left(1 + \frac{R}{100}\right)^T - P$$

  Where:

  - P = Principal amount
  - R = Rate of interest per annum
  - T = Time in years

```python
# Function to calculate simple interest
def calculate_simple_interest(P, R, T):
    SI = (P * R * T) / 100
    return SI

# Function to calculate compound interest
def calculate_compound_interest(P, R, T):
    CI = P * (1 + R / 100) ** T - P
    return CI

# Take input from user
P = float(input("Enter the principal amount: "))
R = float(input("Enter the rate of interest (in %): "))
T = float(input("Enter the time period (in years): "))

# Calculate simple interest
SI = calculate_simple_interest(P, R, T)

# Calculate compound interest
CI = calculate_compound_interest(P, R, T)

# Output the results
print(f"Simple Interest (SI) = {SI}")
print(f"Compound Interest (CI) = {CI}")
```

```
Enter the principal amount: 500
Enter the rate of interest (in %): 2
Enter the time period (in years): 5
Simple Interest (SI) = 50.0
Compound Interest (CI) = 52.040401599999996
```

25. Write a program to get the maximum and minimum values in a dictionary (dict object).

```python
# Sample dictionary
data = {'a': 10, 'b': 25, 'c': 5, 'd': 30}

# Get maximum and minimum values
max_value = max(data.values())
min_value = min(data.values())

print("Maximum value in the dictionary:", max_value)
print("Minimum value in the dictionary:", min_value)
```

```
Maximum value in the dictionary: 30
Minimum value in the dictionary: 5
```