



# Programming in Python File

Name	Parikshit Pandey
Roll Number	05919051723
Branch	IIOT-B1
School	University School of Automation & Robotics
University	Guru Gobind Singh Indraprastha University

# Index

**Q11) Write a Program to check if the input year is a leap year or not**

**Q12) Write a Program to computer factorial of a given number**

**Q13) Write a Program to generate Fibonacci series till 100.**

**Q14) Write a Program to print a multiplication table of a given number**

**Q15) Write a Program to create a two list and perform the following operation's :**

- 1) Add the Elements of the two list.**
- 2) Compare the contents of the two list.**
- 3) to find the number of the elements in the list.**
- 4) Sort the elements of the list**
- 5) Reverse the contents of the List.**

**Q1) Write a Program to perform string manipulation operations using set of pre-defined functions such as :**

- a) Find()**
- b) Upper()**
- c) Len()**
- d) Max() and Min()**
- e) Fetching a specific content from the String**

**Q2) Write a Program to perform to test and check the mathematical functions such as :**

- a) Ceil()**
- b) Sqrt()**
- c) Pow()**
- d) Factorial()**

**Q3) Write a Program that receive a number as input from user and returns if it odd or even number.**

**Q4) Write a Program that receive input from the user to calculate the Area of Triangle**

**Q5) Write a Program that receive input from the user to calculate the Area of Square**

**Q6) Write a Program that receive input from the user to calculate the Area of Rectangle**

**Q7) Write a Program to check if the input string is Palindrome or not**

**Q8) Write a Program that receives marks of a students for a subject as input and assign the grades A||B||C||D||E||F**

**Q9) Write a Program to compute the GCD of the two numbers.**

**Q10) Write a Program to check if the given number is Armstrong number or not. Example of Armstrong number are :- 153, 370, 371 etc.**

- Q16) Write a Program to create and display the content of the tuple. Initialize the tuple with the name of the cities. Display content of the tuple along with name/index positions of the cities.**
- Q17) Write a program to create an Array of Even numbers till 14. Display the contents of array, compute the length of array and also show how to delete a element from the desired position from the array.**
- Q18) Using Filter function, write a program to filter the elements which are greater than 9.**
- Q19) Using Filter function, write a program to display multiple of 5 from a given array.**
- Q20) Write a Program to create a file called "Input.txt", perform the write/read operation in it with a string "Computer Science".**
- Q21) Write a Program to create a file called "Input.txt", initialize it with a string of your choice and perform the read operation to read only the first 3 characters from the file.**
- Q22) Using NumPy, write a program to create 1 Dim Array, load it with numbers, and perform the operation of Iteration and Slicing on it.**
- Q23) Using NumPy, write a program to create Multi-Dim Array, load it with the numbers and display the content of it.**
- Q24) Using NumPy, write a program to create two 1 Dim Array and perform the operation of Iteration, Sorting the contents of array and concatenating the contents of the array.**
- Q25) Using NumPy, initialize the array and display their dimensionality.**
- Q26) Using Panda, create a DataFrame, initialize it with the contents such as : your Enrollment Number and Name and display them.**
- Q27) Create 2 array, using the Matplotlib, plot the graph with the content of the two array, with coordinates plotting on X axis and Y axis.**
- Q28) Create a .csv file(with contents like : Age, Weight and BMI). Read the content of the file and using Panda and Matplotlib, plot the graph.**
- Q29) Create a .csv file(with contents like : Age, Weight and BMI). Read the content of the file and using Panda and Matplotlib, plot the histogram**
- Q30) Write a Program to create a class called 'Student' with fields such as : Enrollment Number, USS Name, Branch Name, Student Name etc. Instantiate a class and make a call to user defined function to display the details of students.**
- Q31) Define Employee Class with fields such as Employee ID and Employee Name. Instantiate the class, invoke the constructor and make a call to user defined function to display the information about employee.**

**Q1) Write a Program to perform string manipulation operations using set of pre-defined functions such as :**

- a) Find()
- b) Upper()
- c) Len()
- d) Max() and Min()
- e) Fetching a specific content from the String

```
def string_operations(input_string):
    print("Input String:", input_string)

    # Find
    print("Index of 'man' in the string:", input_string.find("man"))

    # Upper
    print("Uppercase:", input_string.upper())

    # Len
    print("Length of the string:", len(input_string))

    # Max and Min
    print("Max character:", max(input_string))
    print("Min character:", min(input_string))

    # Fetching specific content
    start_index = input_string.find("brown")
    end_index = input_string.find("jumps") + len("jumps")
    specific_content = input_string[start_index:end_index]
    print("Specific content:", specific_content)

# Example usage
input_string = "The quick brown fox jumps over the lazy dog"
string_operations(input_string)
```

```
Input String: The quick brown fox jumps over the lazy dog
Index of 'man' in the string: -1
Uppercase: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Length of the string: 43
Max character: z
Min character:
Specific content: brown fox jumps
```

**Q2) Write a Program to perform to test and check the mathematical functions such as :**

- a) Ceil()
- b) Sqrt()
- c) Pow()
- d) Factorial()

```
import math

def math_operations(number):
    print("Number:", number)

    # Ceil
    print("Ceiling:", math.ceil(number))

    # Sqrt
    print("Square root:", math.sqrt(number))

    # Pow
    print("Power of 2:", math.pow(number, 2))

    # Factorial
    print("Factorial:", math.factorial(number))

# Example usage
number = float(input("Enter your number : "))
fact = int(input("Enter your factorial number : "))
math_operations(number)
```

```
Enter your number : 14.6
Enter your factorial number : 13
Number: 14.6
Ceiling: 15
Square root: 3.82099463490856
Power of 2: 213.16
Factorial: 6227020800
```

**Q3) Write a Program that receive a number as input from user and returns if it odd or even number.**

```
def check_odd_even():  
    # Get user input  
    number = int(input("Enter a number: "))  
  
    # Check if the number is odd or even  
    if number % 2 == 0:  
        print(f"{number} is an even number.")  
    else:  
        print(f"{number} is an odd number.")  
  
# Call the function to check odd/even  
check_odd_even()
```

```
Enter a number: 56  
56 is an even number.
```

**Q4) Write a Program that receive input from the user to calculate the Area of Triangle**

```
def calculate_triangle_area():  
    # Get base and height from user  
    base = float(input("Enter the base of the triangle: "))  
    height = float(input("Enter the height of the triangle: "))  
  
    # Calculate the area  
    area = 0.5 * base * height  
  
    # Print the result  
    print(f"The area of the triangle with base {base} and height {height} is: {area}")  
  
# Call the function to calculate triangle area  
calculate_triangle_area()
```

```
Enter the base of the triangle: 12  
Enter the height of the triangle: 15  
The area of the triangle with base 12.0 and height 15.0 is: 90.0
```

### Q5) Write a Program that receive input from the user to calculate the Area of Square

```
def calculate_square_area():
    # Get side length from user
    side_length = float(input("Enter the side length of the square: "))

    # Calculate the area
    area = side_length * side_length

    # Print the result
    print(f"The area of the square with side length {side_length} is: {area}")

# Call the function to calculate square area
calculate_square_area()
```

```
Enter the side length of the square: 12
The area of the square with side length 12.0 is: 144.0
```

### Q6) Write a Program that receive input from the user to calculate the Area of Rectangle

```
def calculate_rectangle_area():
    # Get length and width from user
    length = float(input("Enter the length of the rectangle: "))
    width = float(input("Enter the width of the rectangle: "))

    # Calculate the area
    area = length * width

    # Print the result
    print(f"The area of the rectangle with length {length} and width {width} is: {area}")

# Call the function to calculate rectangle area
calculate_rectangle_area()
```

```
Enter the length of the rectangle: 12
Enter the width of the rectangle: 15
The area of the rectangle with length 12.0 and width 15.0 is: 180.0
```

**Q7) Write a Program to check if the input string is Palindrome or not**

```
def is_palindrome(input_string):
    # Remove spaces and convert to lowercase
    input_string = input_string.replace(" ", "").lower()

    # Check if the string is equal to its reverse
    return input_string == input_string[::-1]

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

# Check if it's a palindrome
if is_palindrome(user_input):
    print("The input string is a palindrome.")
else:
    print("The input string is not a palindrome.")
```

```
Enter a string: Bob
The input string is a palindrome.
```

**Q8) Write a Program that receives marks of a students for a subject as input and assign the grades A||B||C||D||E||F**

```
def assign_grade(marks):
    if marks >= 90:
        return "A"
    elif marks >= 80:
        return "B"
    elif marks >= 70:
        return "C"
    elif marks >= 60:
        return "D"
    elif marks >= 50:
        return "E"
    else:
        return "F"

# Get marks from user
marks = float(input("Enter the marks: "))

# Assign grade
grade = assign_grade(marks)

# Print the result
print(f"The grade for {marks} marks is: {grade}")
```

```
Enter the marks: 75
The grade for 75.0 marks is: C
```



**Q9) Write a Program to compute the GCD of the two numbers.**

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

# Get input from user
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

# Compute and print the GCD
print(f"The GCD of {num1} and {num2} is: {gcd(num1, num2)}")
```

```
Enter the first number: 12
Enter the second number: 25
The GCD of 12 and 25 is: 1
```

**Q10) Write a Program to check if the given number is Armstrong number or not. Example of Armstrong number are :- 153, 370, 371 etc.**

```
def is_armstrong_number(number):
    # Calculate the number of digits
    num_digits = len(str(number))

    # Calculate the sum of the nth power of each digit
    sum = 0
    temp = number
    while temp > 0:
        digit = temp % 10
        sum += digit ** num_digits
        temp //= 10

    # Check if the number is an Armstrong number
    return number == sum

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

# Check if it's an Armstrong number
if is_armstrong_number(num):
    print(f"{num} is an Armstrong number.")
else:
    print(f"{num} is not an Armstrong number.")
```

```
Enter a number: 407
407 is an Armstrong number.
```

### Q11) Write a Program to check if the input year is a leap year or not

```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False

# Get input from user
year = int(input("Enter a year: "))

# Check if it's a leap year
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

```
Enter a year: 2020
2020 is a leap year.
```

### Q12) Write a Program to computer factorial of a given number

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

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

# Compute and print the factorial
result = factorial(num)
print(f"The factorial of {num} is: {result}")
```

```
Enter a number: 8
The factorial of 8 is: 40320
```

**Q13) Write a Program to generate Fibonacci series till 100.**

```
def fibonacci_series():
    a, b = 0, 1
    while a <= 100:
        print(a, end=" ")
        a, b = b, a + b

# Generate Fibonacci series
print("Fibonacci series up to 100:")
fibonacci_series()
```

```
Fibonacci series up to 100:
0 1 1 2 3 5 8 13 21 34 55 89
```

**Q14) Write a Program to print a multiplication table of a given number**

```
def multiplication_table(number):
    for i in range(1, 11):
        print(f"{number} x {i} = {number * i}")

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

# Print the multiplication table
print(f"Multiplication table of {num}:")
multiplication_table(num)
```

```
Enter a number: 12
Multiplication table of 12:
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120
```

**Q15) Write a Program to create a two list and perform the following operation's :**

- 1) Add the Elements of the two list.**
- 2) Compare the contents of the two list.**
- 3) to find the number of the elements in the list.**
- 4) Sort the elements of the list**
- 5) Reverse the contents of the List.**

```
def main():
    # Create two lists
    list1 = [1, 2, 3, 4, 5]
    list2 = [6, 7, 8, 9, 10]

    # Add the elements of the two lists
    sum_list = [x + y for x, y in zip(list1, list2)]
    print("Sum of elements of the two lists:", sum_list)

    # Compare the contents of the two lists
    if list1 == list2:
        print("The two lists are equal.")
    else:
        print("The two lists are not equal.")

    # Find the number of elements in the lists
    print("Number of elements in list1:", len(list1))
    print("Number of elements in list2:", len(list2))

    # Sort the elements of the lists
    sorted_list1 = sorted(list1)
    sorted_list2 = sorted(list2)
    print("Sorted list1:", sorted_list1)
    print("Sorted list2:", sorted_list2)

    # Reverse the contents of the lists
    reversed_list1 = list1[::-1]
    reversed_list2 = list2[::-1]
    print("Reversed list1:", reversed_list1)
    print("Reversed list2:", reversed_list2)

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

```
Sum of elements of the two lists: [7, 9, 11, 13, 15]
The two lists are not equal.
Number of elements in list1: 5
Number of elements in list2: 5
Sorted list1: [1, 2, 3, 4, 5]
Sorted list2: [6, 7, 8, 9, 10]
Reversed list1: [5, 4, 3, 2, 1]
Reversed list2: [10, 9, 8, 7, 6]
```

**Q16) Write a Program to create and display the content of the tuple. Initialize the tuple with the name of the cities. Display content of the tuple along with name/index positions of the cities.**

```
def main():
    # Create a tuple with city names
    cities = ("New York", "London", "Paris", "Tokyo", "Sydney")

    # Display the content of the tuple along with index positions
    print("Cities in the tuple:")
    for index, city in enumerate(cities):
        print(f"Index {index}: {city}")

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

```
Cities in the tuple:
Index 0: New York
Index 1: London
Index 2: Paris
Index 3: Tokyo
Index 4: Sydney
```

**Q17) Write a program to create an Array of Even numbers till 14. Display the contents of array, compute the length of array and also show how to delete a element from the desired position from the array.**

```
def main():
    # Create an array of even numbers till 14
    even_numbers = [x for x in range(2, 15, 2)]

    # Display the contents of the array
    print("Array of even numbers till 14:", even_numbers)

    # Compute the length of the array
    print("Length of the array:", len(even_numbers))

    # Delete an element from the desired position (index 2)
    index_to_delete = 2
    if index_to_delete < len(even_numbers):
        del even_numbers[index_to_delete]
        print(f"Array after deleting element at index {index_to_delete}:", even_numbers)
    else:
        print("Index out of range. Element not deleted.")

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

```
Array of even numbers till 14: [2, 4, 6, 8, 10, 12, 14]
Length of the array: 7
Array after deleting element at index 2: [2, 4, 8, 10, 12, 14]
```

**Q18) Using Filter function, write a program to filter the elements which are greater than 9.**

```
def greater_than_9(x):  
    return x > 9  
  
def main():  
    # Create a list of numbers  
    numbers = [1, 10, 5, 8, 12, 3, 9, 15]  
  
    # Use filter function to filter elements greater than 9  
    filtered_numbers = list(filter(greater_than_9, numbers))  
  
    # Print the filtered elements  
    print("Numbers greater than 9:", filtered_numbers)  
  
if __name__ == "__main__":  
    main()
```

Numbers greater than 9: [10, 12, 15]

**Q19) Using Filter function, write a program to display multiple of 5 from a given array.**

```
def is_multiple_of_5(x):  
    return x % 5 == 0  
  
def main():  
    # Create a list of numbers  
    numbers = [2, 5, 10, 15, 7, 20, 25]  
  
    # Use filter function to filter multiples of 5  
    multiples_of_5 = list(filter(is_multiple_of_5, numbers))  
  
    # Print the multiples of 5  
    print("Multiples of 5:", multiples_of_5)  
  
if __name__ == "__main__":  
    main()
```

Multiples of 5: [5, 10, 15, 20, 25]

**Q20) Write a Program to create a file called "Input.txt", perform the write/read operation in it with a string "Computer Science".**

```
def write_to_file(filename, content):
    with open(filename, 'w') as file:
        file.write(content)

def read_from_file(filename):
    with open(filename, 'r') as file:
        content = file.read()
        return content

def main():
    filename = "Input.txt"
    content = "Computer Science"

    # Write to the file
    write_to_file(filename, content)

    # Read from the file
    read_content = read_from_file(filename)
    print("Content read from file:", read_content)

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

Content read from file: Computer Science

**Q21) Write a Program to create a file called "Input.txt", initialize it with a string of your choice and perform the read operation to read only the first 3 characters from the file.**

```
def initialize_file(filename, content):
    with open(filename, 'w') as file:
        file.write(content)

def read_first_n_characters(filename, n):
    with open(filename, 'r') as file:
        content = file.read(n)
        return content

def main():
    filename = "Input.txt"
    content = "Hello, World!"

    # Initialize the file
    initialize_file(filename, content)

    # Read the first 3 characters from the file
    read_content = read_first_n_characters(filename, 3)
    print("First 3 characters read from file:", read_content)

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

First 3 characters read from file: Hel

**Q22) Using NumPy, write a program to create 1 Dim Array, load it with numbers, and perform the operation of Iteration and Slicing on it.**

```
import numpy as np

def main():
    # Create a 1-dimensional NumPy array
    arr = np.array([1, 2, 3, 4, 5])

    # Perform iteration over the array
    print("Iterating over the array:")
    for num in arr:
        print(num)

    # Perform slicing on the array
    print("\nSlicing the array:")
    sliced_arr = arr[1:4]
    print("Sliced array:", sliced_arr)

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

Iterating over the array:

1  
2  
3  
4  
5

Slicing the array:

Sliced array: [2 3 4]

**Q23) Using NumPy, write a program to create Multi-Dim Array, load it with the numbers and display the content of it.**

```
import numpy as np

def main():
    # Create a 2-dimensional NumPy array
    arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

    # Display the content of the array
    print("Multi-dimensional array:")
    print(arr)

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

Multi-dimensional array:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

+ Code

+ Markde



**Q24) Using NumPy, write a program to create two 1 Dim Array and perform the operation of Iteration, Sorting the contents of array and concatenating the contents of the array.**

```
import numpy as np

def main():
    # Create two 1-dimensional NumPy arrays
    arr1 = np.array([3, 1, 2, 5, 4])
    arr2 = np.array([6, 8, 7, 9, 10])

    # Perform iteration over the arrays
    print("Iterating over the first array:")
    for num in arr1:
        print(num)

    # Sort the contents of the arrays
    sorted_arr1 = np.sort(arr1)
    sorted_arr2 = np.sort(arr2)
    print("\nSorted contents of the first array:", sorted_arr1)
    print("Sorted contents of the second array:", sorted_arr2)

    # Concatenate the contents of the arrays
    concatenated_arr = np.concatenate((arr1, arr2))
    print("\nConcatenated array:", concatenated_arr)

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

Iterating over the first array:

3  
1  
2  
5  
4

Sorted contents of the first array: [1 2 3 4 5]

Sorted contents of the second array: [ 6 7 8 9 10]

Concatenated array: [ 3 1 2 5 4 6 8 7 9 10]

**Q25) Using NumPy, initialize the array and display their dimensionality.**

```
import numpy as np

def main():
    # Initialize a NumPy array
    arr = np.array([[1, 2, 3], [4, 5, 6]])

    # Display the dimensionality of the array
    print("Dimensionality of the array:", arr.ndim)

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

Dimensionality of the array: 2

**Q26) Using Panda, create a DataFrame, initialize it with the contents such as : your Enrollment Number and Name and display them.**

```
import pandas as pd

def main():
    # Create a dictionary with enrollment numbers and names
    data = {'Enrollment Number': ['05919051723'],
            'Name': ['Parikshit Pandey']}

    # Create a DataFrame from the dictionary
    df = pd.DataFrame(data)

    # Display the DataFrame
    print("DataFrame:")
    print(df)

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

[9] ✓ 0.0s

```
... DataFrame:
   Enrollment Number      Name
0      05919051723  Parikshit Pandey
```

**Q27) Create 2 array, using the MatPlotLib, plot the graph with the content of the two array, with coordinates plotting on X axis and Y axis.**

```
import matplotlib.pyplot as plt

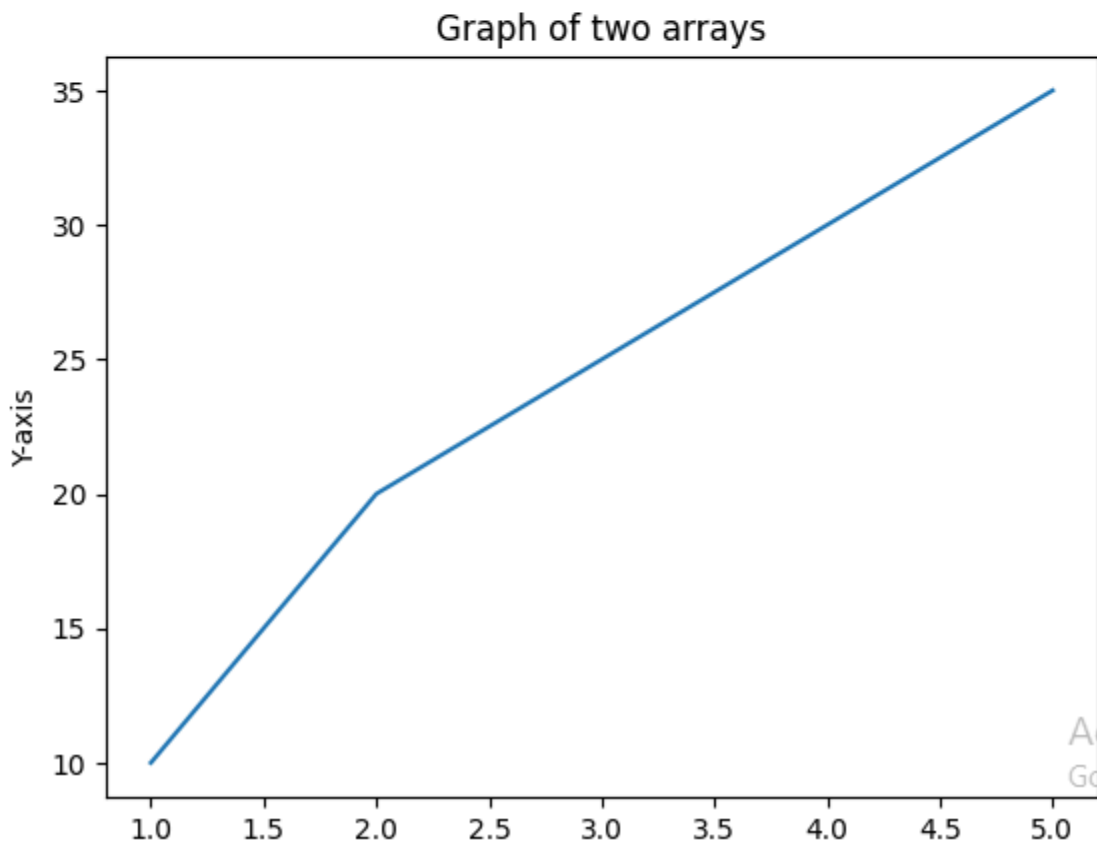
def main():
    # Create two arrays for x and y coordinates
    x = [1, 2, 3, 4, 5]
    y = [10, 20, 25, 30, 35]

    # Plot the graph
    plt.plot(x, y)

    # Add labels and title
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.title('Graph of two arrays')

    # Display the graph
    plt.show()

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



**Q28) Create a .csv file(with contents like : Age, Weight and BMI). Read the content of the file and using Panda and Matplotlib, plot the graph.**

```
import pandas as pd
import matplotlib.pyplot as plt

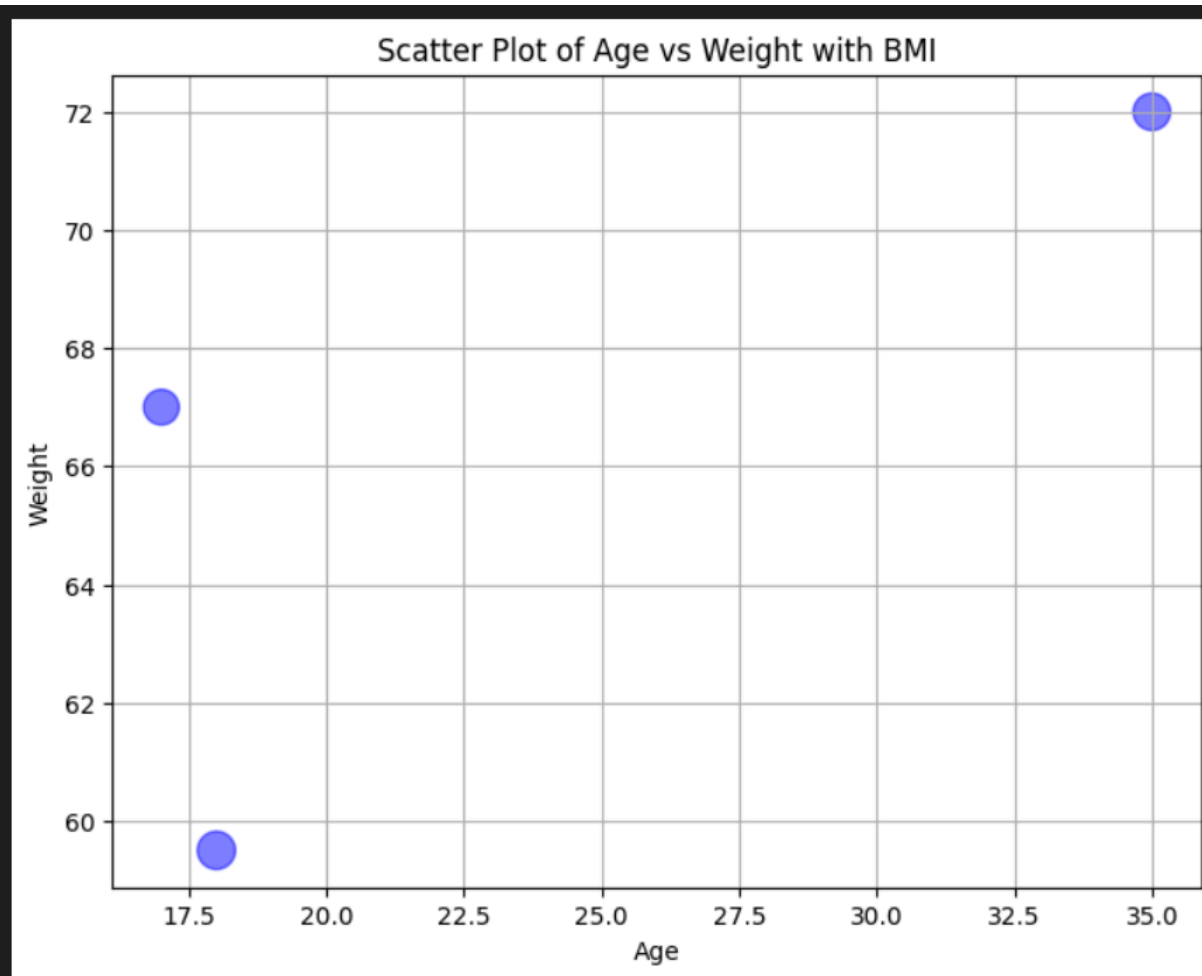
def main():
    # Create a DataFrame with the specified data
    data = {'Age': [18,17,35],
            'Weight': [59.5,67,72],
            'BMI': [24.1,21,23.04]}
    df = pd.DataFrame(data)

    # Write the DataFrame to a CSV file
    df.to_csv('data.csv', index=False)

    # Read the CSV file into a DataFrame
    df_read = pd.read_csv('data.csv')

    # Plot the graph
    plt.figure(figsize=(8, 6))
    plt.scatter(df_read['Age'], df_read['Weight'], s=df_read['BMI']*10, c='blue', alpha=0.5)
    plt.xlabel('Age')
    plt.ylabel('Weight')
    plt.title('Scatter Plot of Age vs Weight with BMI')
    plt.grid(True)
    plt.show()

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



**Q29) Create a .csv file(with contents like : Age, Weight and BMI). Read the content of the file and using Panda and MatPlotLib, plot the histogram**

```
import pandas as pd
import matplotlib.pyplot as plt

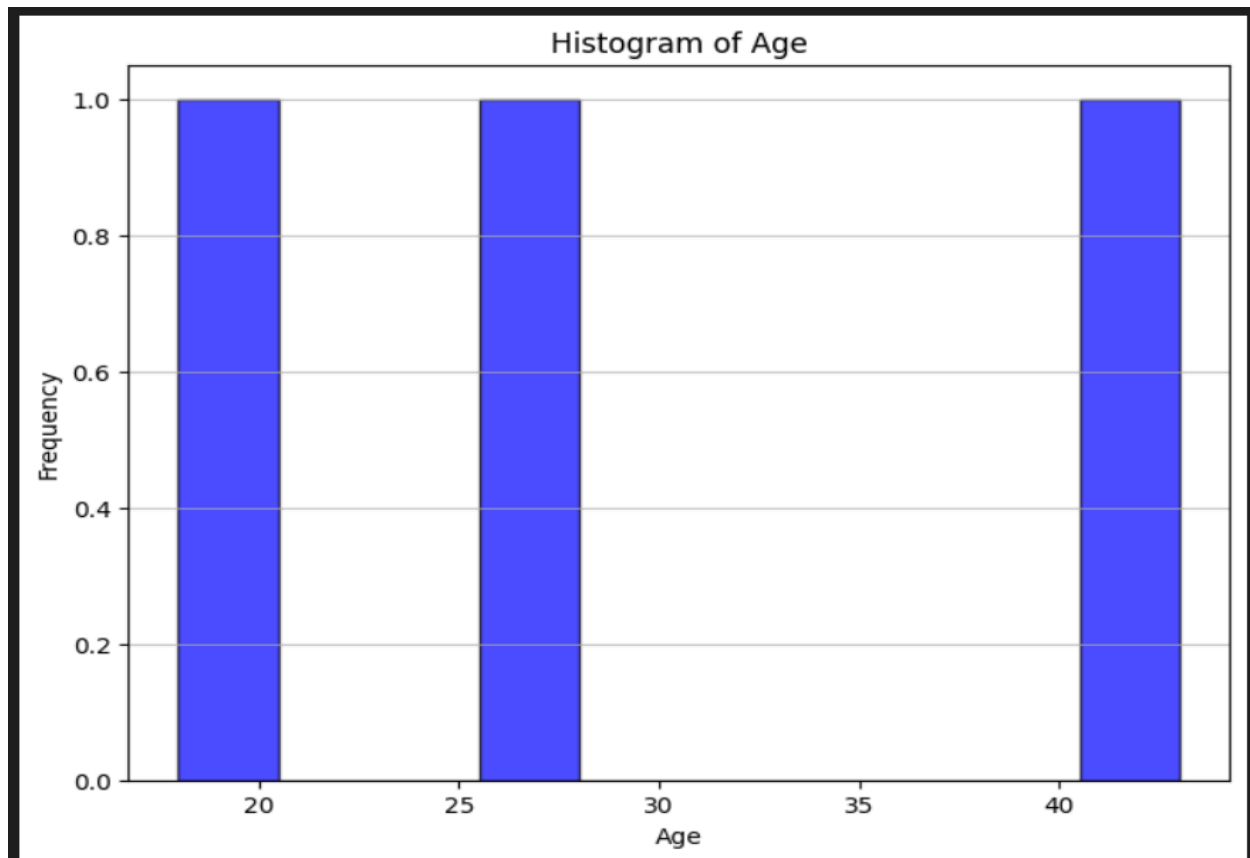
def main():
    # Create a DataFrame with the specified data
    data = {'Age': [18,27,43],
            'Weight': [59.5,65.34,75.3],
            'BMI': [24.1,21.3,24.6]}
    df = pd.DataFrame(data)

    # Write the DataFrame to a CSV file
    df.to_csv('data.csv', index=False)

    # Read the CSV file into a DataFrame
    df_read = pd.read_csv('data.csv')

    # Plot the histogram
    plt.figure(figsize=(8, 6))
    plt.hist(df_read['Age'], bins=10, alpha=0.7, color='blue', edgecolor='black')
    plt.xlabel('Age')
    plt.ylabel('Frequency')
    plt.title('Histogram of Age')
    plt.grid(axis='y', alpha=0.75)
    plt.show()

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



**Q30) Write a Program to create a class called 'Student' with fields such as : Enrollment Number, USS Name, Branch Name, Student Name etc. Instantiate a class and make a call to user defined function to display the details of students.**

```
class Student:
    def __init__(self, enrollment_number, uss_name, branch_name, student_name):
        self.enrollment_number = enrollment_number
        self.uss_name = uss_name
        self.branch_name = branch_name
        self.student_name = student_name

    def display_details(self):
        print("Enrollment Number:", self.enrollment_number)
        print("USS Name:", self.uss_name)
        print("Branch Name:", self.branch_name)
        print("Student Name:", self.student_name)

def main():
    # Instantiate the Student class
    student = Student('05919051723', 'USAR', 'IIOT', 'Parikshit Pandey')

    # Call the display_details method to display student details
    student.display_details()

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

[10] ✓ 0.0s

```
... Enrollment Number: 05919051723
USS Name: USAR
Branch Name: IIOT
Student Name: Parikshit Pandey
```

**Q31) Define Employee Class with fields such as Employee ID and Employee Name. Instantiate the class, invoke the constructor and make a call to user defined function to display the information about employee.**

```
class Employee:
    def __init__(self, employee_id, employee_name):
        self.employee_id = employee_id
        self.employee_name = employee_name

    def display_info(self):
        print("Employee ID:", self.employee_id)
        print("Employee Name:", self.employee_name)

def main():
    # Instantiate the Employee class with specified ID and name
    employee = Employee(59, 'Parikshit Pandey')

    # Call the display_info method to display employee information
    employee.display_info()

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

[11] ✓ 0.0s

```
... Employee ID: 59
Employee Name: Parikshit Pandey
```