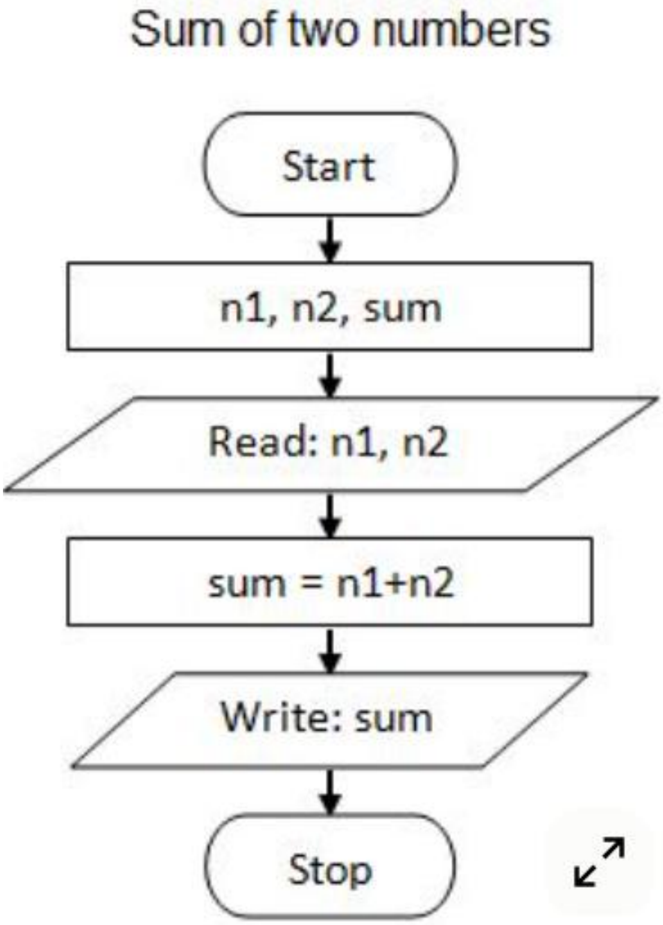
Practical 1

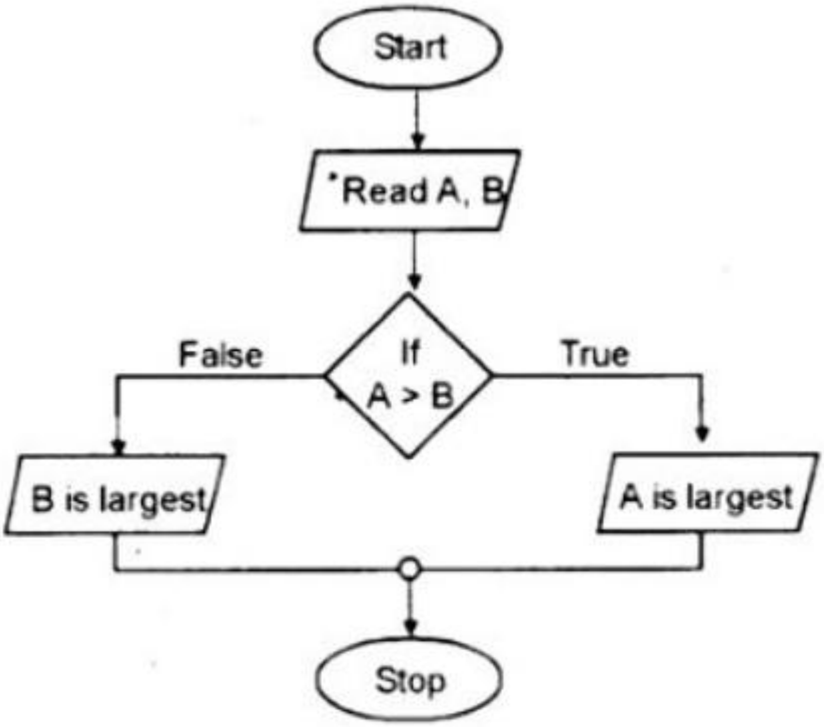
Aim: Prepare flowchart and algorithm for a given problem.

- **Find the sum of two given numbers.**
- **Find a maximum out of two given numbers.**
- **Find whether a given number is odd or even.**
- **Find a maximum out of three given numbers.**

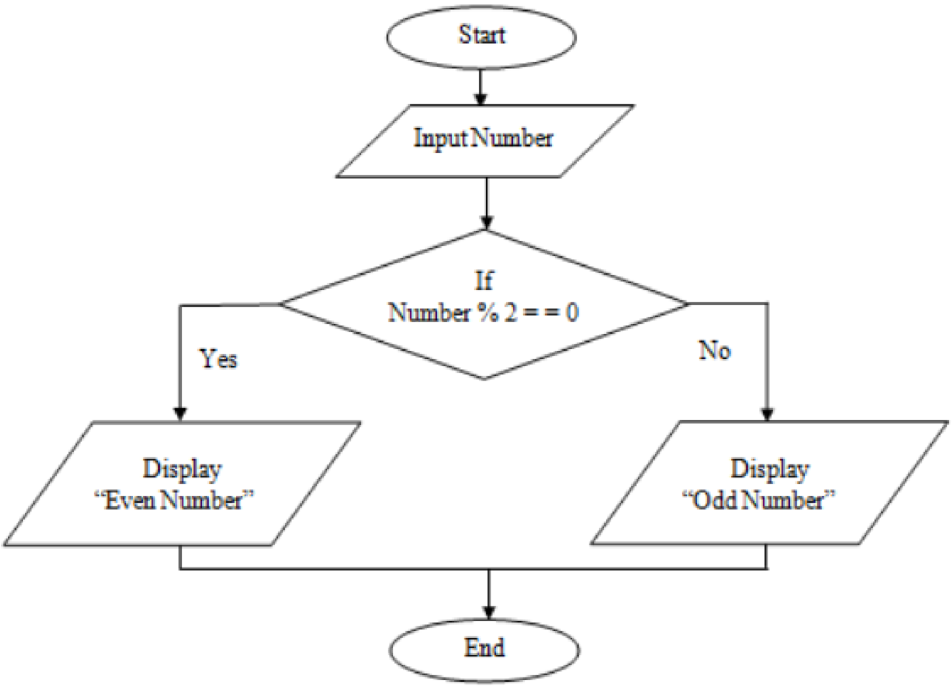
1. **Sum of 2 given numbers**



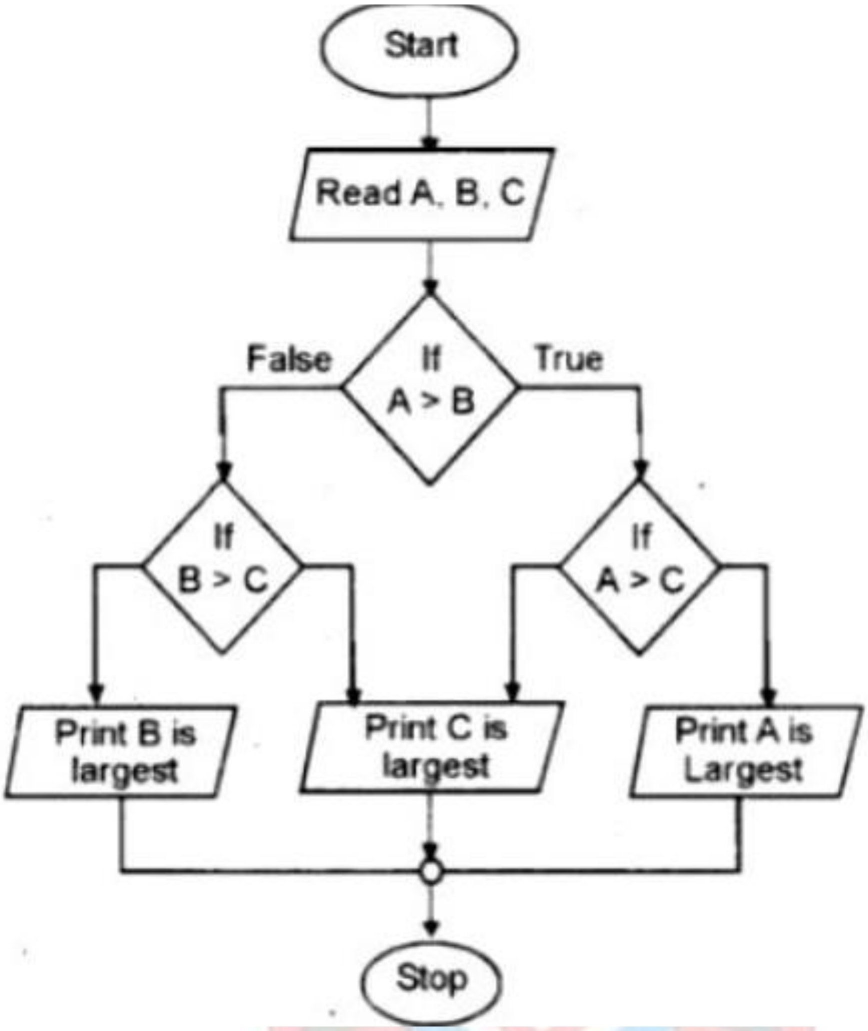
2. **Find max of 2 numbers**



3. **Find if given number is odd or even**



4. Max of 3 numbers



Practical 2

Aim: Install & configure python software and Create a program to print your name, date of birth and mobile number.

Step 1: Install Python

1. Go to the official Python website: https://www.python.org/downloads/
2. Download the latest version for your OS (Windows, Mac, Linux).
3. **Important:** During installation, check the box **"Add Python to PATH"**.
4. Complete the installation.

Step 2: Verify Python Installation

Open your terminal (Command Prompt / PowerShell / Terminal) and type:

```
python --version
```

You should see something like:

```
Python 3.12.0
```

Step 3: Create Your Program

1. Open a text editor (like VS Code, Sublime Text, or Notepad).
2. Save a new file as my_details.py.
3. Add the following Python code:

```
# Program to print personal details

# Variables to store information
name = "John Doe"
date_of_birth = "01-01-2000"
mobile_number = "+1234567890"

# Print the details
print("Name:", name)
print("Date of Birth:", date_of_birth)
print("Mobile Number:", mobile_number)
```

Step 4: Run Your Program

Open terminal, navigate to the folder where my_details.py is saved, and run:

```
python my_details.py
```

Expected Output

```
Name: John Doe
Date of Birth: 01-01-2000
Mobile Number: +1234567890
```

Practical 3

Aim: Develop a program to identify data-types in python.

Code

```
# Example values
name = "Alice"          # String
age = 25                # Integer
height = 5.7            # Float
is_student = True       # Boolean
fruits = ["Apple", "Banana", "Cherry"]  # List

# Function to print data type
def print_data_type(var):
    print(f"The value {var} is of type {type(var)}")

# Check data types
print_data_type(name)
print_data_type(age)
print_data_type(height)
print_data_type(is_student)
print_data_type(fruits)
```

Output

```
The value Alice is of type <class 'str'>
The value 25 is of type <class 'int'>
The value 5.7 is of type <class 'float'>
The value True is of type <class 'bool'>
```

Practical 4

Aim: 1) Create a program to read three numbers from the user and find the average of the numbers.
2) Create a program to convert temperature from Fahrenheit to Celsius unit using eq: C=(F-32)/1.8 .

Code

```
# Read three numbers from the user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

# Calculate average
average = (num1 + num2 + num3) / 3

# Display the result
print("The average of the three numbers is:", average)
```

Output:

```
Enter first number: 10
Enter second number: 20
Enter third number: 30
The average of the three numbers is: 20.0
```

2) Program to Convert Fahrenheit to Celsius

```
# Read temperature in Fahrenheit
fahrenheit = float(input("Enter temperature in Fahrenheit: "))

# Convert to Celsius
celsius = (fahrenheit - 32) / 1.8

# Display the result
print(f"{fahrenheit} Fahrenheit is equal to {celsius:.2f} Celsius")
```

Output:

```
Enter temperature in Fahrenheit: 98.6
98.6 Fahrenheit is equal to 37.00 Celsius
```

‘

Practical 5

Aim: 1) Create a program to identify whether the scanned number is even or odd and print an appropriate message.
2) Create a program to find a maximum number among the given three numbers.

1) Program to Identify Even or Odd Number

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

# Check if the number is even or odd
if num % 2 == 0:
   print(f"{num} is an Even number.")
else:
   print(f"{num} is an Odd number.")
```

Output:

```
Enter a number: 7
7 is an Odd number.
```

2) Program to Find Maximum Among Three Numbers

```
# Read three numbers from the user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

# Find the maximum number
if num1 >= num2 and num1 >= num3:
   maximum = num1
elif num2 >= num1 and num2 >= num3:
   maximum = num2
else:
   maximum = num3

# Display the result
print(f"The maximum number among {num1}, {num2}, and {num3} is {maximum}.")
```

Output:

```
Enter first number: 10
Enter second number: 25
Enter third number: 15
The maximum number among 10.0, 25.0, and 15.0 is 25.0.
```

Practical 6

Aim: Develop a program to show whether the entered number is prime or not.

Code

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

# 0 and 1 are not prime numbers
if num <= 1:
    print(f"{num} is not a prime number.")
else:
    # Assume number is prime
    is_prime = True

    # Check for factors from 2 to num-1
    for i in range(2, num):
        if num % i == 0:
            is_prime = False
            break

    # Display result
    if is_prime:
        print(f"{num} is a prime number.")
    else:
        print(f"{num} is not a prime number.")
```

Output:

```
Enter a number: 11
11 is a prime number.
Enter a number: 12
12 is not a prime number.
```

Practical 7

Aim: Develop a program to print odd and even numbers from 1 to N numbers. (Where N is an integer number entered by the user).

Code.

```
# Read N from the user
N = int(input("Enter the value of N: "))

print("\nEven numbers from 1 to", N, "are:")
for i in range(1, N+1):
    if i % 2 == 0:
        print(i, end=" ")

print("\n\nOdd numbers from 1 to", N, "are:")
for i in range(1, N+1):
    if i % 2 != 0:
        print(i, end=" ")
```

Output:

```
Enter the value of N: 10
Even numbers from 1 to 10 are:
2 4 6 8 10
Odd numbers from 1 to 10 are:
1 3 5 7 9
```

Practical 8

Aim: Develop a program to demonstrate the use of break, continue and pass statements.

Code.

```
for i in range(1, 11):  # Loop from 1 to 10

    if i == 3:
        print("Pass statement at", i)
        pass  # Does nothing, just a placeholder

    if i == 5:
        print("Continue statement at", i)
        continue  # Skip the rest of this iteration

    if i == 8:
        print("Break statement at", i)
        break  # Exit the loop completely

    print("Current number:", i)
```

Output:

```
Current number: 1
Current number: 2
Pass statement at 3
Current number: 3
Current number: 4
Continue statement at 5
Current number: 6
Current number: 7
Break statement at 8
```

Practical 9

Aim: 1) Develop a user-defined function to find the factorial of a given number.

2) Create a user-defined function to print the Fibonacci series of 0 to N numbers. (Where N is an integer number and passed as an argument).

Code:

```
# 1) Factorial of a given number
def factorial(n):
    if n < 0:
        return None  # factorial not defined for negative numbers
    if n == 0 or n == 1:
        return 1
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result


# 2) Fibonacci series up to N numbers
def fibonacci(n):
    if n <= 0:
        return []
    if n == 1:
        return [0]
    series = [0, 1]
    while len(series) < n:
        series.append(series[-1] + series[-2])
    return series


# Example usage:
print("Factorial of 5:", factorial(5))
print("Fibonacci series of 10 terms:", fibonacci(10))
```

Output:

```
Factorial of 5: 120
Fibonacci series of 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Practical 10

Aim: Write a program using the function that reverses the entered value.

Code

```
# Function to reverse entered value
def reverse_value(value):
    value = str(value)  # ensure input is treated as string
    return value[::-1]

# Example usage
print("Reversed string:", reverse_value("Python"))Reversed string: nohtyP
Reversed number: 54321print("Reversed number:", reverse_value(12345))
```

Output:

```
Reversed string: nohtyP
Reversed number: 54321
```

Practical 11

Aim: Write a program that determines whether a given number is an Armstrong number or not using a user-defined function.

Code

```
# Function to check Armstrong number
def is_armstrong(num):
    num_str = str(num)
    power = len(num_str)
    total = sum(int(digit) ** power for digit in num_str)
    return total == num

# Example usage
print("153 is Armstrong:", is_armstrong(153))
print("9474 is Armstrong:", is_armstrong(9474))
print("123 is Armstrong:", is_armstrong(123))
```

Output:

```
153 is Armstrong: True
9474 is Armstrong: True
123 is Armstrong: False
```

Practical 12

Aim: 1) Write a program to reverse words in a given sentence.
2) Write a program to check if a substring is present in a given string.

Code

```
# 1) Reverse words in a given sentence
def reverse_words(sentence):
    return " ".join(sentence.split()[::-1])


# 2) Check if a substring is present in a given string
def contains_substring(string, substring):
    return substring in string

print("Reversed words:", reverse_words("Python is powerful"))
print("Contains 'pro':", contains_substring("Python programming", "pro"))
print("Contains 'java':", contains_substring("Python programming", "java"))
```

Output:

```
Reversed words: powerful is Python
Contains 'pro': True
Contains 'java': False
```

Practical 13

Aim: 1) Create a program to find the sum of all elements in a list using a loop.

2) Create a program to find the smallest and largest element in a given list.

Code

```
# 1) Sum of all elements in a list using loop
def sum_of_list(lst):
    total = 0
    for num in lst:
        total += num
    return total


# 2) Smallest and largest element in a list
def min_max(lst):
    smallest = lst[0]
    largest = lst[0]
    for num in lst:
        if num < smallest:
            smallest = num
        if num > largest:
            largest = num
    return smallest, largest


# Example usage
numbers = [5, 2, 9, 1, 7]

print("Sum of list:", sum_of_list(numbers))
small, large = min_max(numbers)
print("Smallest:", small, "Largest:", large)
```

Output:

```
Sum of list: 24
Smallest: 1 Largest: 9
```

Practical 14

Aim: Given a list saved in variable: a = [1, 8, 7, 15, 25, 36,48, 64, 81, 95]. Write a Python program that takes this list and makes a new list that has only the even elements of this list in it.

Code

```
# Given list
a = [1, 8, 7, 15, 25, 36, 48, 64, 81, 95]

# Extract even elements
even_list = []
for num in a:
    if num % 2 == 0:
        even_list.append(num)

print("Even elements:", even_list)
```

Output:

```
Even elements: [8, 36, 48, 64]
```