

IES College of Technology, Bhopal



Practical File

Name of student :
Enrollment Number :
Subject : Computer Workshop(Python)
Branch : AiMI
Semester : 3rd
Session : 2024-25

LIST OF EXPERIMENTS

S. No	Name of Experiment
1	To write a Python program to find GCD of two numbers?
2	To write a Python Program to find the square root of a number by Newton's Method?
3	WAP to write a Python program to find the exponentiation of a number?
4	WAP to write a Python Program to find the maximum from a list of numbers
5	WAP to write a Python Program to perform Linear Search
6	WAP to write a Python Program to perform binary search
7	WAP to write a Python Program to perform selection sort
8	WAP to write a Python program to find first n prime number
9	WAP to print table of a given number
10	WAP to print factorial of a given number

Experiment No: 1

Objective: To write a Python program to find GCD of two numbers?

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

num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
result = gcd(num1, num2)
print(f"The GCD of {num1} and {num2} is: {result}")
```

Output: The gcd of 60 and 48 is: 12

Experiment No: 2

Objective: To write a Python Program to find the square root of a number by Newton's Method?

```
def sqrt_newton_method(N, tolerance=1e-6, max_iterations=1000):
    x = N / 2
    for _ in range(max_iterations):
        x_new = 0.5 * (x + N / x)
        if abs(x_new - x) < tolerance:
            break
    x = x_new
    return x

num = 327
if num < 0:
    print("Square root is not defined for negative numbers.")
else:
    result = sqrt_newton_method(num)
    print(f"The square root of {num} is approximately: {result}")
```

Output : The square root of 327 is approzimately : 18.0831

Experiment No:3

Objective: WAP to write a Python program to find the exponentiation of a number?

```
def exponentiation(base, exponent):  
    return base ** exponent  
  
base = float(input("Enter the base number: "))  
exponent = float(input("Enter the exponent: "))  
result = exponentiation(base, exponent)  
print(f"The result of {base} raised to the power of {exponent} is: {result}")
```

Output : The result of 2.0 raised to the power of 3.0 is: 8.0

Experiment No:4

Objective: WAP to write a Python Program to find the maximum from a list of numbers

```
# Input: List of numbers  
numbers = [int(x) for x in input("Enter numbers separated by space: ").split()]  
  
# Find the maximum number using the max() function  
max_number = max(numbers)  
  
# Output the result  
print(f"The maximum number is: {max_number}")
```

Output: Enter numbers separated by space: 5 3 9 1 7

The maximum number is: 9

Experiment No:5

Objective: WAP to write a Python Program to perform Linear Search

```
def linear_search(arr, target):  
    for i in range(len(arr)):  
        if arr[i] == target:  
            return i  
    return -1  
  
arr = [10, 20, 30, 40, 50] # Example list  
target = int(input("Enter the number to search for: "))  
result = linear_search(arr, target)  
if result != -1:  
    print(f"Element {target} found at index {result}.")  
else:  
    print(f"Element {target} not found.")
```

Output : Enter the number to search for: 30

Element 30 found at index 2.

Experiment No: 6

Objective: WAP to write a Python Program to perform binary search

```
def binary_search(arr, target):  
    left = 0  
    right = len(arr) - 1  
    while left <= right:  
        mid = (left + right) // 2 # Find the middle index  
        if arr[mid] == target:  
            return mid  
        elif arr[mid] < target:  
            left = mid + 1  
    else:  
        return -1
```

```

        right = mid - 1
    return -1
arr = [10, 20, 30, 40, 50] # Example sorted list
target = int(input("Enter the number to search for: "))
result = binary_search(arr, target)
if result != -1:
    print(f"Element {target} found at index {result}.")
else:
    print(f"Element {target} not found.")

```

Output: Enter the number to search for: 30

Element 30 found at index 2.

Experiment No:7

Objective: WAP to write a Python Program to perform selection sort

```
def selection_sort(arr):
```

```
    n = len(arr)
```

```
    for i in range(n):
```

```
        min_index = i
```

```
        for j in range(i + 1, n):
```

```
            if arr[j] < arr[min_index]:
```

```
                min_index = j
```

```
    element
```

```
    arr[i], arr[min_index] = arr[min_index], arr[i]
```

```
arr = [int(x) for x in input("Enter the list of numbers separated by space: ").split()]
```

```
selection_sort(arr)
```

```
print("Sorted list:", arr)
```

Output: Enter the list of numbers separated by space: 64 25 12 22 11

Sorted list: [11, 12, 22, 25, 64]

Experiment No:8

Objective: WAP to write a Python program to find first n prime number

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num ** 0.5) + 1): # Check divisibility up to the square root of num
        if num % i == 0:
            return False
    return True

def first_n_primes(n):
    primes = []
    num = 2 # Start checking from the first prime number
    while len(primes) < n:
        if is_prime(num):
            primes.append(num)
        num += 1
    return primes

n = int(input("Enter the number of prime numbers to find: "))
primes = first_n_primes(n)
print(f"The first {n} prime numbers are: {primes}")
```

Output: Enter the number of prime numbers to find: 5

The first 5 prime numbers are: [2, 3, 5, 7, 11]

Experiment No: 9

Objective: WAP to print table of a given number

```
def print_table(number, limit=10):  
    for i in range(1, limit + 1):  
        print(f"{number} x {i} = {number * i}")  
  
num = int(input("Enter a number to print its multiplication table: "))  
print_table(num)
```

Output: Enter a number to print its multiplication table: 5

5 x 1 = 5 ,5 x 2 = 10 ,5 x 3 = 15 ,5 x 4 = 20 ,5 x 5 = 25 ,5 x 6 = 30

5 x 7 = 35 ,5 x 8 = 40 ,5 x 9 = 45 ,5 x 10 = 50

Experiment No : 10

Objective : WAP to print factorial of a given number

```
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        fact = 1  
        for i in range(1, n+1):  
            fact *= i  
        return fact  
  
num = int(input("Enter a number to find its factorial: "))  
print(f"The factorial of {num} is {factorial(num)}")
```

Output: Enter a number to find its factorial: 4

The factorial of 4 is 24