POINTER AND FUNCTION

1. Program to Find the Factorial of a Number Using a Function

```
#include <stdio.h>

// Function to calculate factorial
int factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Factorial of %d is %d\n", num, factorial(num));
    return 0;
}
```

2. Program to Find the Greatest of Two Numbers Using a Function

```
#include <stdio.h>

// Function to find the greatest number
int findGreatest(int a, int b) {
    return (a > b) ? a : b;
```

```
}

int main() {
    int num1, num2;

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    printf("The greatest number is %d\n", findGreatest(num1, num2));
    return 0;
}
```

3. Program to Check if a Number is Prime Using a Function

```
#include <stdio.h>
#include <stdbool.h>

// Function to check if a number is prime
bool isPrime(int num) {
    if (num <= 1)
        return false;
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0)
            return false;
    }
    return true;
}

int main() {
    int num;

    printf("Enter a number: ");
```

```
    scanf("%d", &num);

    if (isPrime(num))
        printf("%d is a prime number.\n", num);
    else
        printf("%d is not a prime number.\n", num);

    return 0;
}
```

4. Program to Find the Sum of Elements in an Array Using a Function

```
#include <stdio.h>
// Function to calculate the sum of array elements
int arraySum(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i]
}
    return sum;
}
int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
     printf("The sum of the elements is %d\n", arraySum(arr, n));
return 0;  }
```

5. Recursion

```
#include<stdio.h>
void binary_search(int a[], int low, int high, int key)
{ int mid; mid = (low + high) / 2;
if (low <= high)
 { if (a[mid] == key)
printf("Element found at index: %d\n", mid);
 else if(key < a[mid])
binary_search(a, low, mid-1, key);
 else if (key>a[mid])
 binary_search(a, mid+1, high, key); }
else if (low > high)
printf("Unsuccessful Search\n"); }

 void main()
{ int i, n, low, high, key;
n = 5; low = 0; high = n-1;
int a[10] = {12, 14, 18, 22, 39};
key = 22;
binary_search(a, low, high, key);
 }
```

6. Sorting using pointers

```
#include <stdio.h>
// Function to sort the numbers using pointers
void sort(int n, int* ptr)
{
    int i, j, t;
    // Sort the numbers using pointers
    for (i = 0; i < n; i++) {
```

```
        for (j = i + 1; j < n; j++) {

            if (*(ptr + j) < *(ptr + i)) {

                t = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = t;
            }
        }
    }
    // print the numbers
    for (i = 0; i < n; i++)
        printf("%d ", *(ptr + i));
}
  void main()
{
    int n = 5;
    int arr[] = { 0, 23, 14, 12, 9 };
    sort(n, arr);
}
```