

## 1. Factorial of a Number

```
#include <stdio.h>

factorial(int n) {
    if(n == 0) return 1;
    return n * factorial(n - 1);
}

void main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Factorial = %d\n", factorial(num));
}
```

Input: 5

Output: Factorial = 120

---

## 2. Check Prime Number

```
#include <stdio.h>

int isPrime(int n) {
    if (n <= 1) return 0;
    for (int i = 2; i <= n/2; i++)
        if (n % i == 0) return 0;
    return 1;
}

void 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);  
}
```

Input: 7

Output: 7 is a prime number.

---

### 3. Power Using Recursion

```
#include <stdio.h>  
int power(int base, int exp) {  
    if (exp == 0) return 1;  
    return base * power(base, exp - 1);  
}  
void main() {  
    int b, e;  
    printf("Enter base and exponent: ");  
    scanf("%d %d", &b, &e);  
    printf("Result = %d\n", power(b, e));  
}
```

Input: 2 3

Output: Result = 8

---

### 4. Palindrome Using Recursion

```
#include <stdio.h>  
int isPalindrome(int n, int rev) {  
    if (n == 0) return rev;  
    return isPalindrome(n / 10, rev * 10 + n % 10);  
}
```

```

void main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (num == isPalindrome(num, 0))
        printf("Palindrome number.\n");
    else
        printf("Not a palindrome.\n");
}

```

Input: 121

Output: Palindrome number.

---

## 5. Calculate nCr

```

#include <stdio.h>
int fact(int n) {
    if (n == 0) return 1;
    return n * fact(n - 1);
}
int nCr(int n, int r) {
    return fact(n) / (fact(r) * fact(n - r));
}
void main() {
    int n, r;
    printf("Enter n and r: ");
    scanf("%d %d", &n, &r);
    printf("nCr = %d\n", nCr(n, r));
}

```

Input: 5 2

Output: nCr = 10

---

## 6. Call by Value and Call by Reference

```
#include <stdio.h>
void callByValue(int a) {
    a = a + 10;
}
void callByReference(int *b) {
    *b = *b + 10;
}
void main() {
    int x = 5, y = 5;
    callByValue(x);
    callByReference(&y);
    printf("Call by Value: %d\n", x);
    printf("Call by Reference: %d\n", y);
}
```

Output:

Call by Value: 5

Call by Reference: 15

---

## 7. Swap Two Numbers Using Function

```
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
void main() {  
    int x, y;  
    printf("Enter two numbers: ");  
    scanf("%d %d", &x, &y);  
    swap(&x, &y);  
    printf("After swap: x = %d, y = %d\n", x, y);  
}
```

Input: 3 4

Output: After swap: x = 4, y = 3

---

## 8. Recursive Fibonacci

```
#include <stdio.h>  
int fib(int n) {  
    if (n <= 1) return n;  
    return fib(n - 1) + fib(n - 2);  
}  
void main() {  
    int n;  
    printf("Enter n: ");  
    scanf("%d", &n);  
    printf("%dth Fibonacci = %d\n", n, fib(n));  
}
```

Input: 6

Output: 6th Fibonacci = 8

---

## 9. GCD and LCM Using Functions

```

#include <stdio.h>
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
void lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}
int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("GCD = %d\n", gcd(x, y));
    printf("LCM = %d\n", lcm(x, y));
}

```

Input: 12 18

Output:

GCD = 6

LCM = 36

---

## 10. Global and Local Variables

```

#include <stdio.h>
int globalVar = 10;

void display() {
    int localVar = 5;
    printf("Local variable = %d\n", localVar);
    printf("Global variable = %d\n", globalVar);
}

void main() {

```

```
    display();  
    globalVar = 20;  
    display();  
}
```

Output:

```
Local variable = 5  
Global variable = 10  
Local variable = 5  
Global variable = 20
```