Recursion

1. print n to 1 using recursion

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
5 4 3 2 1
```

> Source Code

```c
#include <stdio.h>

int main(){
    void printN(int);
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    printN(n);
    return 0;
}

void printN(int n){
    if(n == 0){
        return;
    }
    printf("%d ", n);
    printN(n - 1);
}
```

2. print using 1 to n using recursion.

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
1 2 3 4 5
```

> Source Code

```c
#include <stdio.h>

int main(){
    void printN(int);
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    printN(n);
    return 0;
}

void printN(int n){
    if(n == 0){
        return;
    }
    printN(n - 1);
    printf("%d ", n);
}
```

3. Factorial of a number using recursion.

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
Factorial of 5 = 120
```

> Source Code

```c
#include <stdio.h>

int main(){
    int fact(int);
    int n, f;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    f = fact(n);
```

```
    printf("Factorial of %d = %d", n, f);
    return 0;
}
int fact(int n){
    if(n == 0)
        return 1;
    return n * fact(n - 1);
}
```

4. Addition of two numbers using recursion

> Test Data

```
Enter two numbers: 10 5
```

> Expected Output

```
Sum = 15
```

> Source Code

```
#include <stdio.h>

int main(){
    int sum(int, int);
    int n1, n2, s;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    s = sum(n1, n2);

    printf("Sum = %d", s);
    return 0;
}

// Function Defination
int sum(int n1, int n2){
    if(n2 == 0){    // Base Case
        return n1;
    }
    sum(n1 + 1, n2 - 1);  // Recursive Case
}
```

5. Subtraction of two numbers using recursion.

> Test Data

```
Enter two numbers: 20 5
```

> Expected Output

```
Sum = 15
```

> Source Code

```c
#include <stdio.h>

int main(){
    int sub(int, int);
    int n1, n2, s;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    s = sub(n1, n2);

    printf("Subtraction = %d", s);
    return 0;
}

int sub(int n1, int n2){
    if(n2 == 0){
        return n1;
    }
    sub(n1 - 1, n2 - 1);
}
```

6. Multiplication of two numbers using recursion.

> Test Data

```
Enter two numbers: 20 5
```

> Expected Output

```
Sum = 100
```

> Source Code

```c
#include <stdio.h>

int main(){
    int mul(int, int);
    int n1, n2, m;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    m = mul(n1, n2);

    printf("Multiplication = %d", m);
    return 0;
}
```

```
int mul(int n1, int n2){
    if(n2 == 1){
        return n1;
    }
    return n1 + mul(n1, n2 - 1);
}
```

7. Division of two numbers using recursion.

> Test Data

```
Enter two numbers: 20 5
```

> Expected Output

```
Sum = 4
```

> Source Code

```
#include <stdio.h>

int main(){
    int div(int, int);
    int n1, n2, d;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    d = div(n1, n2);

    printf("Division = %d", d);
    return 0;
}
int div(int n1, int n2){
    if(n1 == 0){
        return 0;
    }
    return 1 + div(n1 - n2, n2);
}
```

8. Recursive Program to print multiplication table of a number

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
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
```

> Source Code

```c
#include <stdio.h>

int main(){
    void printTable(int, int);
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    printTable(n, 1);
    return 0;
}

void printTable(int n, int t){
    if(t > 10){
        return;
    }
    printf("\n%d X %d = %d",n, t, n * t);
    printTable(n, t + 1);
}
```

9. C program to calculate power of a number using recursion.

> Test Data

```
Enter the value of base and power: 3 4
```

> Expected Output

```
3 to the power of 4 = 81
```

> Source Code

```c
#include <stdio.h>

int main(){
    int power(int, int);
    int x, y, p;

    printf("Enter the value of base and power: ");
    scanf("%d%d", &x, &y);

    p = power(x, y);
```

```
    printf("%d to the power %d = %d", x, y, p);
    return 0;
}

int power(int x, int y){
    if(y == 0)
        return 1;
    return x * power(x, y - 1);
}
```

10. C program to count digits of a number using recursion.

> Test Data

```
Enter the value of n: 12345
```

> Expected Output

```
Total number of digits = 5
```

> Source Code

```
#include <stdio.h>

int main(){
    int countNum(int);
    int n, c;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    c = countNum(n);

    printf("Total number of digits = %d", c);
    return 0;
}

int countNum(int n){
    if(n == 0){
        return 0;
    }
    return 1 + countNum(n / 10);
}
```

11. C program to find sum of all digits using recursion.

> Test Data

```
Enter the value of n: 12345
```

> Expected Output

```
Sum of digits = 15
```

> Source Code

```c
#include <stdio.h>

int main(){
    int sum(int);
    int n, s;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    s = sum(n);

    printf("Sum of digits = %d", s);
    return 0;
}

int sum(int n){
    if(n == 0)
        return 0;
    return n % 10 + sum(n / 10);
}
```

12. C program to reverse an integer number using recursion

> Test Data

```
Enter the value of n: 123
```

> Expected Output

```
Reverse number = 321
```

> Source Code

```c
#include <stdio.h>
#include <math.h>

int main(){
    int reverse(int, int);
    int num, length = 0, temp, rev;

    printf("Enter the value of n: ");
    scanf("%d", &num);

    temp = num;
    while(temp > 0){
        length++;
        temp /= 10;
    }
    rev = reverse(num, length - 1);
```

```
    printf("Reverse number = %d", rev);
    return 0;
}

int reverse(int n, int length){
    if(length == 0)
        return n;

    return (n % 10) * pow(10, length) + reverse(n / 10, length - 1);
}
```

13. C program to check a given number is prime or not using recursion

> Test Data

```
Enter the value of n: 15
```

> Expected Output

```
Not a prime number
```

> Source Code

```
#include <stdio.h>

int main(){
    int isPrime(int, int);
    int num;

    printf("Enter the value of n: ");
    scanf("%d", &num);

    if(isPrime(num, num / 2)){
        printf("Prime number");
    } else{
        printf("Not a prime number");
    }
    return 0;
}

int isPrime(int n, int i){
    if(i == 1)
        return 1;
    if(n % i == 0)
        return 0;
    isPrime(n, --i);
}
```

14. C program to find the HCF (Highest Common Factor) of given numbers using recursion

> Test Data

```
Enter two numbers: 10 20
```

> Expected Output

```
The highest common factor is: 10
```

> Source Code

```c
#include <stdio.h>

int main(){
    int hcf(int, int);
    int n1, n2, h;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    h = hcf(n1, n2);

    printf("The highest common factor is: %d", h);
    return 0;
}

int hcf(int n1, int n2){
    if(n2 != 0)
        return hcf(n2, n1 % n2);
    else
        return n1;
}
```

15. Write a program in C to print the array elements using recursion.

> Expected Output

```
1 2 3 4 5
```

> Source Code

```c
#include <stdio.h>

int main(){
    void printArray(int [], int);
    int arr[5] = {1, 2, 3, 4, 5};
    int length = 5;

    printArray(arr, length - 1);

    return 0;
}

void printArray(int arr[], int length){
    if(length < 0)
        return ;
    printArray(arr, length - 1);
```

```
    printf("%d ", arr[length]);
}
```

16. Write a program in C to get the largest element of an array using recursion.

> Array

```
{5, 7, 8, 9, 6}
```

> Expected Output

```
max = 9
```

> Source Code

```
#include <stdio.h>

int main(){
    int findMax(int [], int, int);
    int arr[5] = {5, 7, 8, 9, 6};
    int length = 5, max;

    max = findMax(arr, length - 1, 0);

    printf("Max = %d", max);
    return 0;
}

int findMax(int arr[], int length, int max){
    if(length < 0)
        return max;
    if(max < arr[length])
        max = arr[length];
    findMax(arr, length - 1, max);
}
```

17. Write a program in C to convert a decimal number to binary using recursion

> Test Data

```
Enter a number: 10
```

> Expected Output

```
Binary number = 1010
```

> Source Code

```c
#include <stdio.h>

int main(){
    int dec_bin(int);
    int num, bin;

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

    bin = dec_bin(num);

    printf("Binary number = %d", bin);
    return 0;
}

int dec_bin(int num){
    if(num == 0)
        return 0;
    return num % 2 + 10 * dec_bin(num / 2);
}
```

18. Write a program in C to find the LCM of two numbers using recursion.

> Test Data

```
Enter two numbers: 125 162
```

> Expected Output

```
LCM = 1134
```

> Source Code

```c
#include <stdio.h>

int main(){
    int lcm(int, int, int);
    int n1, n2, l, max;

    printf("Enter two numbers: ");
    scanf("%d%d",&n1,&n2);

    max = n1;
    if(max < n2)
        max = n2;
    l = lcm(n1, n2, max);

    printf("LCM = %d", l);
    return 0;
}

int lcm(int n1, int n2, int max){
    if(max % n1 == 0 && max % n2 == 0)
        return max;
```

```
    lcm(n1, n2, max+1);
}
```

19. Write a program in C to print even or odd numbers in a given range using recursion.

> Test Data

```
Input the range to print starting from 1: 10
```

> Expected Output

```
2 4 6 8 10
```

> Source Code

```
#include <stdio.h>

int main(){
    void printEven(int);
    int n;

    printf("Input the range to print starting from 1: ");
    scanf("%d", &n);

    printEven(n);
    return 0;
}

void printEven(int n){
    if(n == 1)
        return ;
    printEven(n-1);
    if(n % 2 == 0)
        printf("%d ", n);
}
```

20. Write a C program to check whether a given number is a palindrome or not using recursion.

> Test Data

```
Enter any number: 4224
```

> Expected Output

```
Palindrome number
```

> Source Code

```c
#include <stdio.h>

int main(){
    int isPalindrome(int, int);
    int num, rev = 0;

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

    if(isPalindrome(num, rev) == num)
        printf("Palindrome number");
    else
        printf("Not a palindrome number");

    return 0;
}

int isPalindrome(int num, int rev){
    if(num == 0)
        return rev;
    isPalindrome(num / 10, rev * 10 + (num % 10));
}
```

Hi 👋, I'm Rajiv Kumar

A passionate frontend developer from India

🌐 Connect with me:

GITHUB TWITTER LINKEDIN

💻 Tech Stack: