Loop related problem:

1. Write a C program to print all-natural numbers from 1 to n and their sum.

```
#include<stdio.h>
int main() {
    int n, sum = 0;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        printf("%d ", i);
        sum = sum + i;
    }
    printf("\nSum : %d", sum);
}
```

2. Write a C program to print all natural numbers in reverse order(n to 1)
3. Write a C program to calculate summation of even numbers and odd numbers between 1 to n.

```
#include<stdio.h>
int main() {
    int n, even_sum = 0, odd_sum = 0;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        if(i%2 == 0) {
            even_sum += i;
        } else {
            odd_sum += i;
        }
    }
    printf("Sumation of even number between 1 to %d is %d\n",n,even_sum);
    printf("Sumation of odd number between 1 to %d is %d\n",n,odd_sum);
}
```

4. Summation of this series: 1+1/2+ 1/3+…..... + 1/n

```
#include<stdio.h>
int main() {
    int n;
    float sum = 0;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        sum = sum + (1.0) / i;
    }
    printf("Sum = %.2f", sum);
}
```

5. Summation of this series: 1^2+2^2+3^2+……+n^2

6. Summation of this series: 1^2-4^2+7^2-10^2……..+n^2

```
#include<stdio.h>
int main() {
    int n, sum = 0;
    scanf("%d", &n);

    for(int i=1; i<=n; i+=3) {
        if(i%2==0) {
            sum = sum - (i*i);
        } else {
            sum = sum + (i*i);
        }
    }
    printf("Sum = %d", sum);
}
```

7. Summation of this series: 1+(1+2) +(1+2+3)+……+(1+2+3…+n)

```
#include<stdio.h>
int main() {
    int n, sum = 0;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        sum = sum + (i*(i+1))/2;
    }
    printf("Sum : %d", sum);
}
```

8. Summation of this series: 1^2+3^2/3^3 +5^2/5^3 ……..

9. Write a C program to print multiplication table of any number.

10. Generate a multiplication table of all the numbers from 1 to n.

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=10; j++) {
         printf("%d * %d = %d\n", i, j, i*j);
        }
        printf("\n");
    }
}
```

11. Find the factorial of a given number.

```
#include<stdio.h>
int main() {
    int n, fact = 1;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        fact = fact * i;
    }
    printf("Factorial of %d is %d", n, fact);
}
```

12. Calculate npr and ncr value.

13. Check whether a given number is prime or not.

```
#include<stdio.h>
int main() {
    int n, isPrime = 1;
    scanf("%d", &n);

     if(n==1) isPrime = 0;

     for(int i=2; i<n; i++) {
        if(n % i == 0) {
         isPrime = 0;
         break;
        }
    }
    if(isPrime) printf("%d is prime number!", n);
    else printf("%d is not prime number!", n);
}
```

14. Write a C program to print all Prime numbers between 1 to n.

15. Calculate the power of a given number using a loop.

```
#include<stdio.h>
int main() {
    int base, power, result = 1;
    scanf("%d %d", &base, &power);

    for(int i=1; i<=power; i++) {
     result = result * base;
    }

    printf("%d to the power %d is %d", base, power, result);
}
```

16. Write a C program to find all factors of a number.

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    printf("Factors of %d : ", n);
    for(int i=1; i<=n; i++) {
        if(n%i == 0) {
            printf("%d ", i);
        }
    }
}
```

17. Find out prime factor of a number.

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        if(n % i == 0) {
            int flag = 1;
            if(i==1) flag = 0;
            for(int j=2; j<i; j++) {
                if(i % j == 0) {
                    flag = 0;
                    break;
                }
            }
            if(flag) printf("%d ", i);
        }
    }
}
```

18. Print the Fibonacci sequence up to a given number

```
#include<stdio.h>
int main() {
    int n, first = 0, sec = 1, third = first + sec;
    scanf("%d", &n);

    printf("%d %d ", first, sec);

    while(third <= n) {
        printf("%d ", third);
        first = sec;
        sec = third;
        third = first + sec;
    }
}
```

19. Find the GCD and LCM of two number.

```
#include <stdio.h>
int main() {
    int m, n;
    scanf("%d %d", &m, &n);
    while (n) {
        int r = m % n;
        m = n;
        n = r;
    }
    printf("GCD = %d \n",m);
    printf("LCM = %d \n",(m*n) / GCD);
}
```

20. Find the sum of all the digits of a given number and count digit in a number.

```
#include<stdio.h>
int main() {
    int n, sum = 0, total_dig = 0;;
    scanf("%d", &n);

    while(n) {
        sum = sum + (n%10);
        n = n / 10;
        total_dig++;
    }

    printf("Total Digit : %d\n", total_dig);
    printf("Sumation of Digit : %d", sum);
}
```

21. Write a C program to check whether a number is Armstrong number or not.

An Armstrong number is a number such that sum of its digits raised to nth power is equal to the number itself.

```
#include<stdio.h>
#include<math.h>
int main() {
    int n, total_dig = 0, orginial_num;
    float result = 0.0;
    scanf("%d", &n);
    orginial_num = n;

    while(n) {
        n = n / 10;
        total_dig++;
    }
    n = orginial_num;
    while(n) {
        result = result + pow(n%10, total_dig);
        n = n / 10;
    }
    if((int)result == orginial_num)
        printf("%d is Armstrong number!", orginial_num);
    else
        printf("%d is not Armstrong number!", orginial_num);
}
```

22. Write a C program to check whether a number is Perfect number or not.

A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself.

```
#include<stdio.h>
#include<math.h>
int main() {
    int n, sum = 0;
    scanf("%d", &n);
    for(int i=1; i<n; i++) {
        if(n % i == 0) {
            sum += i;
        }
    }
    if(sum == n)
        printf("%d is perfect number!", n);
    else
        printf("%d is not perfect number!", n);
}
```

23. Write a C program to check whether a number is Strong number or not.

Strong number is a number whose sum of all digits' factorial is equal to the number 'n'.

24. Reverse a number and check a number palindrome or not

```
#include <stdio.h>

int main() {
  int n, reverse = 0, remainder;
  scanf("%d", &n);

  while (n) {
    remainder = n % 10;
    reverse = reverse * 10 + remainder;
    n /= 10;
  }

  printf("Reversed number = %d", reverse);
}
```

25. Write a C program to convert Binary to Decimal number system.

```
#include <stdio.h>
int main()
{
    int num, binary_num, decimal_num = 0, base = 1, rem;
    scanf("%d", &num);

    binary_num = num;

    while (num){
        rem = num % 10;
        decimal_num = decimal_num + rem * base;
        num = num / 10;
        base = base * 2;
    }

    printf("The binary number is %d", binary_num);
    printf("\nThe decimal number is %d", decimal_num);
}
```

Function:

1. Summation of 1 to n using recursion.

```
#include <stdio.h>
int sum(int n) {
    if (n == 1)
        return 1;
    else
        return n + sum(n - 1);
}

int main() {
    int n;
    scanf("%d", &n);
    printf("Sum of natural numbers from 1 to %d is %d", n, sum(n));
}
```

2. Find the factorial of N using recursion.

```
#include <stdio.h>
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}
int main() {
    int n;
    scanf("%d", &n);
    printf("Factorial of %d is %d", n, factorial(n));
}
```

3. Find the n^{th} Fibonacci number and print the Fibonacci number series.

```
#include <stdio.h>
int fibonacci(int n) {
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fibonacci(n-1) + fibonacci(n-2);
}

int main() {
    int n;
    scanf("%d", &n);
    printf("The %dth Fibonacci number is %d", n, fibonacci(n));
}
```

4. Count the digits of a number using recursion.

```
#include <stdio.h>
int countDigits(int n) {
    if (n == 0)
        return 0;
    else
        return 1 + countDigits(n/10);
}

int main() {
    int n;
    scanf("%d", &n);
    printf("Number of digits in %d is %d", n, countDigits(n));
}
```

5. Find the sum of the digit using recursion.

```
#include <stdio.h>
int sumDigits(int n) {
    if (n == 0)
        return 0;
    else
        return (n % 10) + sumDigits(n/10);
}

int main() {
      int n;
      scanf("%d", &n);
      printf("Sum of digits in %d is %d", n, sumDigits(n));
}
```

6. Reverse a number using recursion.

```
#include <stdio.h>
int reverseNumber(int n) {
    static int reversedNum = 0;
    if (n == 0)
        return 0;
    else {
        reversedNum = reversedNum*10 + n%10;
        reverseNumber(n/10);
    }
    return reversedNum;
}

int main() {
    int n;
    scanf("%d", &n);
    printf("The reversed number is %d", reverseNumber(n));
}
```

7. Find GCD using recursion.

```
#include <stdio.h>
int gcd(int n1, int n2) {
    if (n2 == 0)
        return n1;
    else
        return gcd(n2, n1 % n2);
}

int main() {
    int n1, n2;
    scanf("%d %d", &n1, &n2);
    printf("The GCD of %d and %d is %d", n1, n2, gcd(n1, n2));
}
```

Array:

1. Find the maximum and minimum elements in an array.

```
#include <stdio.h>
int main() {
    int arr[10] = {5, 2, 8, 1, 9, 3, 6, 4, 7, 0};
    int max = arr[0], min = arr[0];

    for (int i = 1; i < 10; i++) {
        if (arr[i] > max) max = arr[i];
        if (arr[i] < min) min = arr[i];
    }
    printf("Maximum element: %d\n", max);
    printf("Minimum element: %d\n", min);
}
```

2. Find the summation of all elements in an array.

```
#include <stdio.h>
int main() {
    int arr[10] = {5, 2, 8, 1, 9, 3, 6, 4, 7, 0};
    int sum = 0;

    for (int i = 0; i < 10; i++) {
        sum += arr[i];
    }
    printf("Sumation of all elements : %d", sum);
}
```

8. Write a program for transpose of a matrix.

```
#include<stdio.h>
int main() {
    int rows, cols;
    scanf("%d %d", &rows, &cols);
    int A[10][10], transpose[10][10];

    for(int i=0; i<rows; i++) {
        for(int j=0; j<cols; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    for(int i=0; i<rows; i++) {
        for(int j=0; j<cols; j++) {
            transpose[j][i] = A[i][j];
        }
    }

    printf("Transpose Matrix : \n");
    for(int i=0; i<cols; i++) {
        for(int j=0; j<rows; j++) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }
}
```

9. Write a program find the sum of diagonal elements of a matrix

```
#include<stdio.h>
int main() {
    int A[][3] = {
        {10, 20, 30},
        {40, 50, 60},
        {70, 80, 90}
    };

    int sum = 0;
    for(int i=0; i<3; i++) {
        for(int j=0; j<3; j++) {
            if(i==j) sum += A[i][j];
        }
    }
    printf("Sum of diagonal elements : %d", sum);
}
```

String:

1. Write a program to find the length of a string without using library functions.

```
#include <stdio.h>
int main() {
    char str[100];
    int l= 0;

    gets(str);

    while(str[l]!='\0') {
        l++;
    }
    printf("Length of the string is : %d", l);
}
```

2. Write a program to count the total number of words in a string.

```
#include <stdio.h>
int main() {
    char str[100];
    int i = 0, total_word = 1;

    gets(str);

    while(str[i]!='\0') {
        if(str[i]==' ') {
            total_word++;
        }
        i++;
    }
    printf("Total Words in '%s' : %d", str, total_word);
}
```

3. Write a program in C to count the total number of alphabets, digits and special characters in a string.

```
#include <stdio.h>
int main() {
    char str[100];
    gets(str);
    int alp = 0, digit = 0, splch = 0, i;
    while(str[i]!='\0')
    {
        if((str[i]>='a'&&str[i]<='z')||(str[i]>='A'&&str[i]<='Z')) {
            alp++;
        } else if(str[i]>='0' && str[i]<='9') {
            digit++;
        }
        else {
            splch++;
        }
        i++;
    }
    printf("Number of Alphabets in the string is : %d\n", alp);
    printf("Number of Digits in the string is : %d\n", digit);
    printf("Number of Special characters in the string is : %d", splch);
}
```

4. Write a program to concatenate two string without using library function.

```
#include <stdio.h>
int main() {
    char str1[100], str2[100];
    int i = 0, j = 0;
    gets(str1);
    gets(str2);

    while(str1[i] != '\0') i++;

    while(str2[j] != '\0')  {
        str1[i] = str2[j];
        j++;
        i++;
    }
    str1[i] = '\0';
    printf("\nConcatenated String is %s", str1);
}
```

5. Write a program in C to read a sentence and replace lowercase characters with uppercase and vice versa.

```
#include <stdio.h>
int main() {
    char str[100];
    gets(str);

    int len = 0;
    while(str[len] != '\0') {
        len++;
    }

    for(int i=0; i<len; i++) {
        if(str[i] >= 65 && str[i] <= 90) {
            str[i] = str[i] + 32;
        } else if(str[i] >= 97 && str[i] <= 122) {
            str[i] = str[i] - 32;
        }
    }
    printf("%s", str);
}
```

6. Write a program to reverse a string and check palindrome or not.

```
#include <stdio.h>
int main() {
    char str[100];
    gets(str);

    int len = 0, flag = 1;
    while(str[len] != '\0') {
        len++;
    }

    for(int i=0, j=len-1; i<len/2; i++, j--){
        if(str[i] != str[j]) {
            flag = 0;
            break;
        }
    }
    if(flag) {
        printf("%s is palindrom string!", str);
    } else {
        printf("%s is not palindrom string!", str);
    }
}
```

(i) * * * * * * * * * *	(ii) * * * * * * * * * *	(iii) * * * * * * * * * *	(iv) * * * * * * * * * *
(v) * * * * * * * * * * * * * * * *	(vi) 1 1 2 1 2 3 1 2 3 4	(vii) 4 3 2 1 3 2 1 2 1 1	(viii) 1 1 2 1 2 3 1 2 3 4
(ix) 4 3 2 1 3 2 1 2 1 1	(x) 4 4 4 4 3 3 3 2 2 1	(xi) 1 2 1 3 2 1 4 3 2 1	(xii) 1 2 4 3 6 9 4 8 12 16
(xiii) 1 1 0 1 0 1 1 0 1 0	(xiv) 1 0 1 1 0 1 0 1 0 1	(xv) AABBAA AABB AA	(xvi) AA AABB AABBAA
(xvii) 1 2 3 1 2 1 1 2 1 2 3	(xviii) * * * * * * * * * * * *		

i)
```c
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=i; j++) {
            printf("*");
        }
        printf("\n");
    }
}
```

```
*
* *
* * *
* * * *
```

ii)
```c
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=n; i>=1; i--) {
        for(int j=1; j<=i; j++) {
            printf("*");
        }
        printf("\n");
    }
}
```

```
* * * *
* * *
* *
*
```

iii)
```c
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=(n-i); j++) {
            printf(" ");
        }
        for(int j=1; j<=i; j++) {
            printf("*");
        }
        printf("\n");
    }
}
```

```
    *
   * *
  * * *
 * * * *
```

iv) Try Yourself

v)
```c
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=(n-i); j++) {
            printf(" ");
        }
        for(int j=1; j<=(i*2-1); j++) {
            printf("*");
        }
        printf("\n");
    }
}
```

```
    *
   * * *
  * * * * *
 * * * * * * *
```

vi) Try Yourself

vii)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=n; i>=1; i--) {
        for(int j=i; j>=1; j--) {
            printf("%d", j);
        }
        printf("\n");
    }
}
```

```
4 3 2 1
3 2 1
2 1
1
```

viii) Try Yourself

ix) Try Yourself

x)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=n; i>=1; i--) {
        for(int j=1; j<=i; j++) {
            printf("%d", i-1+1);
        }
        printf("\n");
    }
}
```

```
4 4 4 4
3 3 3
2 2
1
```

xi) Try Yourself

xii)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=i; j++) {
            printf("%d", i*j);
        }
        printf("\n");
    }
}
```

```
1
2 4
3 6 9
4 8 12 16
```

xiii)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=1; j<=i; j++) {
            if(j%2) printf("1");
            else printf("0");
        }
        printf("\n");
    }
}
```

```
1
1 0
1 0 1
1 0 1 0
```

xiv)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=1; i<=n; i++) {
        for(int j=i; j>=1; j--) {
            if(j%2) printf("1");
            else printf("0");
        }
        printf("\n");
    }
}
```

```
1
0 1
1 0 1
0 1 0 1
```

xv)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=n; i>=1; i--) {
        for(int j=1; j<=i; j++) {
            if(j%2) printf("AA");
            else printf("BB");
        }
        printf("\n");
    }
}
```

```
AABBAA
AABB
AA
```

xvi) Try Yourself

xvii)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for(int i=n; i>=1; i--) {
        for(int j=1; j<=i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
    for(int i=2; i<=n; i++) {
        for(int j=1; j<=i; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
}
```

```
1 2 3
1 2
1
1 2
1 2 3
```

xviii)

```
#include<stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    for(int i=1; i<=n; i++) {
        for(int j=1; j<=n; j++) {
            if(i==1 || i==n || j==1 || j==n)
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
}
```

```
* * * *
*     *
*     *
* * * *
```