Factorial

```
#include <stdio.h>
unsigned long long int factorial (unsigned int i)
{
    if(i <= 1){

        return 1;
    }

    return i * factorial (i - 1);


}

int main(){

    int i = 24;

    printf("factorial of %u is %lld \n" , i , -1 * (factorial(i)));
    return 0;


}
```

Fibonacci Series recurtion

```
#include <stdio.h>
int fibonacci(int i)
{
    if (i==0)
    {
        return 0;
    }




    if (i==1)
    {
        return 1;
    }
    return fibonacci(i-1)+fibonacci(i-2);
}
int main ()
{
```

```
    int i;
    printf ("enter the last num");
    scanf("%d" , &i);

    for(int j = 0 ; j < i ; j++ ){
        printf ("%d \t \n", fibonacci(j));
    }

    return 0;
}
```

Count Digits

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

}

int main(){
int n;
printf ("enter a num");
scanf("%d", &n);
{
    if (n==0)
    printf ("number doesnt exist");
    else
    printf ("the number of  digits are: %d", cd(n));

    return 0;

}

}
```

Prime Factors

```
#include <stdio.h>

// Function to find the smallest prime factor of n
int myfactor(int n) {
    if (n % 2 == 0) {
        return 2;
    }
```

```
    for (int i = 3; i <= n; i += 2) {
        if (n % i == 0) {
            return i;
        }
    }
    return n;
}

// Function to print all prime factors of n
void allprime(int n) {
    if (n == 1) {
        return;
    }
    int factor = myfactor(n);
    printf("%d ", factor);
    allprime(n / factor);
}

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

    printf("Prime factors: ");
    allprime(num);
    printf("\n");

    return 0;
}
```

Krishnamurthy Number

```
#include <stdio.h>

// Function to calculate factorial of a digit
int factorial(int n) {
    int fact = 1;
    for(int i = 1; i <= n; i++) {
        fact *= i;
    }
    return fact;
}

// Function to check Krishnamurthy number
int isKrishnamurthy(int num) {
    int original = num;
    int sum = 0;
```

```
    while(num > 0) {
        int digit = num % 10;
        sum += factorial(digit);
        num /= 10;
    }

    return sum == original;
}

int main() {
    int number;

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

    if(isKrishnamurthy(number)) {
        printf("%d is a Krishnamurthy number.\n", number);
    } else {
        printf("%d is not a Krishnamurthy number.\n", number);
    }

    return 0;
}
```

1+2+3…n using Recursion

```
#include <stdio.h>

int sum(int n) {
    if (n == 0)
        return 0;
    else
        return n + sum(n - 1);
}

int main() {
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    printf("Sum = %d\n", sum(n));
    return 0;
}
```

GCD

```
#include <stdio.h>

int gcd(int a, int b) {
    if (b == 0)
        return a;
    else
        return gcd(b, a % b);
}

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("GCD = %d\n", gcd(x, y));
    return 0;
}
```

Finding an element

```
#include <stdio.h>
int main ()
{

    int flag=0, key ,temp ;
    printf("Enter the size of array:");
    scanf("%d", &key);


    int x[key];
    for (int i=0;  i<key; i++) {

        scanf("%d" , &x[i]);

    }

    printf("What u want to find?");
    scanf("%d" , &temp);

    for(int j=0 ;j < key ; j++){
```

```
        if (x[j]==temp){

        printf("elements found at position %d",j);
        flag=1;
        break;
        }
    }


    if(flag == 0){

        printf("Sorry , not found");
    }

    return 0;
}
```

Max n Min in Array

```
#include <stdio.h>

int main() {
    int n, i;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d numbers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int max = arr[0], min = arr[0];

    for(i = 1; i < n; i++) {
        if(arr[i] > max)
            max = arr[i];
        if(arr[i] < min)
            min = arr[i];
    }

    printf("Maximum: %d\n", max);
    printf("Minimum: %d\n", min);

    return 0;
}
```

Even Odd in Array

```
#include <stdio.h>

int main() {
    int n, i;
    int numbers[100], even[100], odd[100];
    int evenCount = 0, oddCount = 0;

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

    printf("Enter %d numbers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &numbers[i]);

        if(numbers[i] % 2 == 0) {
            even[evenCount++] = numbers[i];
        } else {
            odd[oddCount++] = numbers[i];
        }
    }

    printf("\nEven numbers (%d): ", evenCount);
    for(i = 0; i < evenCount; i++) {
        printf("%d ", even[i]);
    }

    printf("\nOdd numbers (%d): ", oddCount);
    for(i = 0; i < oddCount; i++) {
        printf("%d ", odd[i]);
    }

    printf("\n");

    return 0;
}
```

Add Two Matrices

```
#include <stdio.h>

int main() {
    int rows, cols, i, j;
    int matrix1[50][50], matrix2[50][50], sum[50][50];

    printf("Enter the number of rows: ");
```

```
    scanf("%d", &rows);
    printf("Enter the number of columns: ");
    scanf("%d", &cols);

    // Input matrix1
    printf("Enter elements of first matrix (%d x %d):\n", rows, cols);
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    // Input matrix2
    printf("Enter elements of second matrix (%d x %d):\n", rows, cols);
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }

    // Add matrices
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            sum[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

    // Display result
    printf("Resultant matrix after addition:\n");
    for(i = 0; i < rows; i++) {
        for(j = 0; j < cols; j++) {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Multiply Two Matrices

```
#include <stdio.h>

int main() {
    int r1, c1, r2, c2, i, j, k;
    int matrix1[50][50], matrix2[50][50], product[50][50];
```

```
// Input dimensions
printf("Enter rows and columns of first matrix: ");
scanf("%d%d", &r1, &c1);

printf("Enter rows and columns of second matrix: ");
scanf("%d%d", &r2, &c2);

// Check if multiplication is possible
if (c1 != r2) {
   printf("Matrix multiplication not possible!\n");
   return 1;
}

// Input first matrix
printf("Enter elements of first matrix:\n");
for(i = 0; i < r1; i++) {
   for(j = 0; j < c1; j++) {
      scanf("%d", &matrix1[i][j]);
   }
}

// Input second matrix
printf("Enter elements of second matrix:\n");
for(i = 0; i < r2; i++) {
   for(j = 0; j < c2; j++) {
      scanf("%d", &matrix2[i][j]);
   }
}

// Initialize product matrix to 0
for(i = 0; i < r1; i++) {
   for(j = 0; j < c2; j++) {
      product[i][j] = 0;
   }
}

// Matrix multiplication
for(i = 0; i < r1; i++) {
   for(j = 0; j < c2; j++) {
      for(k = 0; k < c1; k++) {
         product[i][j] += matrix1[i][k] * matrix2[k][j];
      }
   }
}

// Display result
printf("Product of the matrices:\n");
```

```
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            printf("%d ", product[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Digit into Words

```
            #include <stdio.h>

void printDigitInWords(int digit) {
    switch(digit) {
        case 0: printf("Zero "); break;
        case 1: printf("One "); break;
        case 2: printf("Two "); break;
        case 3: printf("Three "); break;
        case 4: printf("Four "); break;
        case 5: printf("Five "); break;
        case 6: printf("Six "); break;
        case 7: printf("Seven "); break;
        case 8: printf("Eight "); break;
        case 9: printf("Nine "); break;
    }
}

int main() {
    int num, reversed = 0;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (num == 0) {
        printf("Zero\n");
        return 0;
    }

    // Reverse the number to maintain the left-to-right order
    int temp = num;
    while(temp > 0) {
        reversed = reversed * 10 + temp % 10;
        temp /= 10;
    }

    // Print digits in words
```

```
    while(reversed > 0) {
        int digit = reversed % 10;
        printDigitInWords(digit);
        reversed /= 10;
    }

    printf("\n");
    return 0;
}
```

Changing case

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

int main() {
    char str[100];
    int i;

    printf("Enter a string: ");
    gets(str);

    for(i = 0; str[i] != '\0'; i++) {
        if(isupper(str[i]))
            str[i] = tolower(str[i]);
        else if(islower(str[i]))
            str[i] = toupper(str[i]);
    }

    printf("Altered case string: %s", str);

    return 0;
}
```

number of vowels, consonants, spaces, and special characters

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

int main() {
    char str[200];
    int vowels = 0, consonants = 0, spaces = 0, special = 0;

    printf("Enter a sentence: ");
    gets(str);
```

```
    for(int i = 0; str[i] != '\0'; i++) {
        char ch = str[i];

        if(isalpha(ch)) {
            ch = tolower(ch);
            if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                vowels++;
            else
                consonants++;
        }
        else if(ch == ' ')
            spaces++;
        else if(ch != '\n')  // ignore newline from fgets
            special++;
    }

    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);
    printf("Spaces: %d\n", spaces);
    printf("Special Characters: %d\n", special);

    return 0;
}
```

string palindrome

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char str[100], cleanStr[100];
    int i, j = 0, len, isPalindrome = 1;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    // Remove non-alphanumeric characters and convert to lowercase
    for(i = 0; str[i] != '\0'; i++) {
        if(isalnum(str[i])) {
            cleanStr[j++] = tolower(str[i]);
        }
    }
    cleanStr[j] = '\0';
    len = j;
```

```
    // Check for palindrome
    for(i = 0; i < len / 2; i++) {
        if(cleanStr[i] != cleanStr[len - i - 1]) {
            isPalindrome = 0;
            break;
        }
    }

    if(isPalindrome)
        printf("The string is a palindrome.\n");
    else
        printf("The string is not a palindrome.\n");

    return 0;
}
```

String count

```
#include <stdio.h>

int mylen(char str[]){
    int i=0;
    while(str[i] != '\0'){
        i += 1;
    }
    return i;

}



int main() {
    char myStr[30];
    printf("\n enter a string :");
    gets(myStr);
    printf("The length of the string is : %d", mylen(myStr));


    return 0;
}
```

pointer counting

```
#include <stdio.h>

int stringLength(char *str) {
    int length = 0;
    while (*str != '\0') {
        length++;
        str++;
    }
    return length;
}

int main() {
    char str[100];
    printf("Enter a string: ");
    gets(str);

    printf("Length of the string: %d\n", stringLength(str));
    return 0;
}
```

1d array using malloc()

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

int main() {
    int *arr;
    int n, i;

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

    // Allocate memory for n integers
    arr = (int *)malloc(n * sizeof(int));

    // Check if memory allocation was successful
    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    // Input elements
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
```

```
    // Display elements
    printf("The elements are:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    // Free allocated memory
    free(arr);

    return 0;
}
```

Copying a file

```
#include <stdio.h>

int main() {
    FILE *src, *dest;
    char ch;
    char sourceName[100], targetName[100];

    printf("Enter source file name: ");
    scanf("%s", sourceName);

    printf("Enter target file name: ");
    scanf("%s", targetName);

    src = fopen(sourceName, "r");
    dest = fopen(targetName, "w");

    if (!src || !dest) {
        printf("Error opening files.\n");
        return 1;
    }

    while ((ch = fgetc(src)) != EOF)
        fputc(ch, dest);

    printf("File copied successfully.\n");

    fclose(src);
    fclose(dest);
    return 0;
}
```

Count characters, words, and lines in a text file.

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

int main() {
    FILE *fp;
    char filename[100], ch;
    int charCount = 0, wordCount = 0, lineCount = 0;
    int inWord = 0;

    printf("Enter the file name: ");
    scanf("%s", filename);

    fp = fopen(filename, "r");

    if (fp == NULL) {
        printf("File cannot be opened.\n");
        return 1;
    }

    while ((ch = fgetc(fp)) != EOF) {
        charCount++;

        if (ch == '\n')
            lineCount++;

        if (isspace(ch)) {
            inWord = 0;
        } else if (inWord == 0) {
            inWord = 1;
            wordCount++;
        }
    }

    fclose(fp);

    printf("Characters: %d\n", charCount);
    printf("Words: %d\n", wordCount);
    printf("Lines: %d\n", lineCount);

    return 0;
}
```

1. Linear Search

```
#include <stdio.h>

int main() {
    int a[100], n, i, key;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);

    printf("Enter the key to search: ");
    scanf("%d", &key);

    for(i = 0; i < n; i++) {
        if(a[i] == key) {
            printf("Element found at position %d\n", i + 1);
            return 0;
        }
    }

    printf("Element not found.\n");
    return 0;
}
```

2. Binary Search (array must be sorted)

```
#include <stdio.h>

int main() {
    int a[100], n, key, low, high, mid, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d sorted elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);

    printf("Enter the key to search: ");
    scanf("%d", &key);
```

```
    low = 0;
    high = n - 1;

    while(low <= high) {
        mid = (low + high) / 2;
        if(a[mid] == key) {
            printf("Element found at position %d\n", mid + 1);
            return 0;
        }
        else if(a[mid] < key)
            low = mid + 1;
        else
            high = mid - 1;
    }

    printf("Element not found.\n");
    return 0;
}
```

Bubble sort

```
#include <stdio.h>

int main() {
    int a[100], n, i, j, temp;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);

    for(i = 0; i < n - 1; i++) {
        for(j = 0; j < n - i - 1; j++) {
            if(a[j] > a[j + 1]) {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }

    printf("Sorted array: ");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
```

```
}
```

4. Selection Sort

```
#include <stdio.h>

int main() {
    int a[100], n, i, j, min, temp;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);

    for(i = 0; i < n - 1; i++) {
        min = i;
        for(j = i + 1; j < n; j++) {
            if(a[j] < a[min])
                min = j;
        }
        if(min != i) {
            temp = a[i];
            a[i] = a[min];
            a[min] = temp;
        }
    }

    printf("Sorted array: ");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
}
```

Insertion sort

```
#include <stdio.h>

int main() {
    int a[100], n, i, j, key;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
```

```
        scanf("%d", &a[i]);

    for(i = 1; i < n; i++) {
        key = a[i];
        j = i - 1;

        while(j >= 0 && a[j] > key) {
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = key;
    }

    printf("Sorted array: ");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
}
```

two complex numbers and display their sum and difference using structure

```
#include <stdio.h>

typedef struct {
    float real;
    float imag;
} Complex;

int main() {
    Complex c1, c2, sum, diff;

    // Input
    printf("Enter first complex number (real and imaginary): ");
    scanf("%f %f", &c1.real, &c1.imag);

    printf("Enter second complex number (real and imaginary): ");
    scanf("%f %f", &c2.real, &c2.imag);

    // Sum
    sum.real = c1.real + c2.real;
    sum.imag = c1.imag + c2.imag;

    // Difference
    diff.real = c1.real - c2.real;
    diff.imag = c1.imag - c2.imag;

    // Output
```

```
    printf("\nSum = %.2f + %.2fi\n", sum.real, sum.imag);
    printf("Difference = %.2f + %.2fi\n", diff.real, diff.imag);

    return 0;
}
```