

```
// SWAP 2 NUMBERS USING FUNCTION
```

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

```
int swap(int a, int b)
```

```
{
```

```
    int temp;
```

```
    temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    printf("After swapping: %d, %d", a, b);
```

```
}
```

```
int main()
```

```
{
```

```
    int x, y;
```

```
    printf("Enter 2 numbes: ");
```

```
    scanf("%d %d", &x, &y);
```

```
    printf("Before swapping: %d, %d\n", x, y);
```

```
    swap(x,y);
```

```
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//SORT AN ARRAY USING FUNCTION
```

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

```
// Function to perform bubble sort
```

```
void bubbleSort(int arr[], int n)
```

```
{
```

```
    int i, j, temp;
```

```
    for (i = 0; i < n-1; i++)
```

```
    {
```

```
        for (j = 0; j < n-i-1; j++)
```

```
        {
```

```
            // Swap if the element found is greater than the next element
```

```
            if (arr[j] > arr[j+1])
```

```
            {
```

```
                temp = arr[j];
```

```
                arr[j] = arr[j+1];
```

```
                arr[j+1] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```

{
    int i, n;
    int arr[50];
    printf("Enter number of items: ");
    scanf("%d", &n);

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

    printf("Unsorted array: \n");
    for(i=0; i<n; i++)
    {
        printf("%d ", arr[i]);
    }

    // Call the sorting function
    bubbleSort(arr, n);

    printf("\nSorted array: \n");
    for(i=0; i<n; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}

```

```

////////////////////////////////////

```

```

//MATRIX ADDITION

```

```

#include <stdio.h>
void main()
{
    int r, c;
    int arr1[50][50], arr2[50][50], arr3[50][50];
    int i, j;

    printf("Enter number of Rows: ");
    scanf("%d", &r);

    printf("Enter number of Columns: ");
    scanf("%d", &c);

```

```

/*****

```

```

printf("\nEnter elements for Matrix-1: ");
for(i=0; i<r; i++)
{
    printf("\nEnter elements for Row-%d\n", i+1);
    for(j=0; j<c; j++)
    {
        printf("Column-%d: ", j+1);
        scanf("%d", &arr1[i][j]);
    }
}

```

```

printf("\nElements for Matrix-1: \n\n");
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        printf("%d ", arr1[i][j]);
    }
    printf("\n\n");
}

```

/******

```

printf("\nEnter elements for Matrix-2: ");
for(i=0; i<r; i++)
{
    printf("\nEnter elements for Row-%d\n", i+1);
    for(j=0; j<c; j++)
    {
        printf("Column-%d: ", j+1);
        scanf("%d", &arr2[i][j]);
    }
}

```

```

printf("\nElements for Matrix-2: \n\n");
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        printf("%d ", arr2[i][j]);
    }
    printf("\n\n");
}

```

/******

```

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        arr3[i][j] = arr1[i][j] + arr2[i][j];
    }
}

/*****/

printf("After Matrix Addition: \n\n");
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
    {
        printf("%d ", arr3[i][j]);
    }
    printf("\n\n");
}

/////////////////////////////////////////////////////////////////

//FIBONACCI SERIES UPTO N
#include <stdio.h>
int main()
{
    int n, first = 0, second = 1, next;

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

    printf("Fibonacci Series up to %d terms: \n", n);

    // Print the first two terms
    printf("%d %d ", first, second);

    // Generate the Fibonacci series
    for (int i = 3; i <= n; i++)
    {
        next = first + second;
        printf("%d ", next);

        // Update first and second for the next iteration

```

```

        first = second;
        second = next;
    }

    printf("\n");

    return 0;
}

////////////////////////////////////

//READ A STRING AND CHECK WHETHER IT IS PALINDROME OR NOT
#include <stdio.h>
#include <string.h>

#define MAX_LENGTH 100
int main()
{
    char str[MAX_LENGTH];
    int i, len, isPalindrome = 1; // Assume it's a palindrome initially

    // Input a string from the user
    printf("Enter a string: ");
    scanf("%s", str);

    // Calculate the length of the string
    len = strlen(str);

    // Check if the string is a palindrome
    for (i = 0; i < len / 2; i++)
    {
        if (str[i] != str[len - i - 1])
        {
            isPalindrome = 0; // It's not a palindrome
            break;
        }
    }

    // Output the result
    if (isPalindrome)
    {
        printf("%s is a palindrome.\n", str);
    }
    else
    {
        printf("%s is not a palindrome.\n", str);
    }
}

```

```

    }

    return 0;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//MATRIX ADDITION
#include <stdio.h>
void main()
{
    int r1, c1, r2, c2;
    int arr1[50][50], arr2[50][50], arr3[50][50];
    int i, j;

    printf("Enter rows and columns for the first matrix: ");
    scanf("%d %d", &r1, &c1);

    printf("\nEnter elements for Matrix-1: ");
    for(i=0; i<r1; i++)
    {
        printf("\nEnter elements for Row-%d\n", i+1);
        for(j=0; j<c1; j++)
        {
            printf("Column-%d: ", j+1);
            scanf("%d", &arr1[i][j]);
        }
    }

    printf("\n\nEnter rows and columns for the second matrix: ");
    scanf("%d %d", &r2, &c2);

    // Check if matrices can be multiplied
    if (c1 != r2)
    {
        printf("Error! Number of columns in the first matrix should be equal to th
    }
    else
    {
        printf("\nEnter elements for Matrix-2: ");
        for(i=0; i<r2; i++)
        {
            printf("\nEnter elements for Row-%d\n", i+1);
            for(j=0; j<c2; j++)
            {

```

```

        printf("Column-%d: ", j+1);
        scanf("%d", &arr2[i][j]);
    }
}

// Initializing elements of the result matrix to 0
for(int i = 0; i < r1; ++i)
{
    for(int j = 0; j < c2; ++j)
    {
        arr3[i][j] = 0;
    }
}

// Multiplying firstMatrix and secondMatrix and storing the result in resultMatrix
for(int i = 0; i < r1; ++i)
{
    for(int j = 0; j < c2; ++j)
    {
        for(int k = 0; k < c1; ++k)
        {
            arr3[i][j] = arr3[i][j] + (arr1[i][k] * arr2[k][j]);
        }
    }
}

// Display the result
printf("\nResultant Matrix:\n");
for(int i = 0; i < r1; ++i)
{
    for(int j = 0; j < c2; ++j)
    {
        printf("%d ", arr3[i][j]);
    }
    printf("\n\n");
}
}

```

////////////////////////////////////

//CHECK WHETHER GIVEN NUMBER IS ARMSTRONG OR NOT

```

#include<stdio.h>
int main()
{
    int n,r,sum=0,temp;

```

```

printf("Enter the number: ");
scanf("%d",&n);
temp=n;

while(n>0)
{
    r=n%10;
    sum=sum+(r*r*r);
    n=n/10;
}

if(temp==sum)
    printf("Armstrong number ");
else
    printf("Not an Armstrong number");
return 0;
}

////////////////////////////////////

//READ N INTEGERS IN AN ARRAY AND SORT ELEMENTS
#include <stdio.h>
int main()
{
    int n, arr[50];

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

    // Input elements into the array
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    // Sorting the array in ascending order (using Bubble Sort)
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                // Swap elements if they are in the wrong order
                int temp = arr[j];

```



```

        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
    }
}

// Display the sorted array
printf("Sorted array in ascending order:\n");
for (int i = 0; i < n; i++)
{
    printf("%d ", arr[i]);
}
printf("\n");

return 0;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//ADD 2 NUMBERS USING POINTERS
#include <stdio.h>
int main()
{
    int num1, num2, sum;
    int *ptr1, *ptr2;

    // Input the first number
    printf("Enter the first number: ");
    scanf("%d", &num1);

    // Input the second number
    printf("Enter the second number: ");
    scanf("%d", &num2);

    // Pointers pointing to the addresses of num1 and num2
    ptr1 = &num1;
    ptr2 = &num2;

    // Adding two numbers using pointers
    sum = *ptr1 + *ptr2;

    // Display the result
    printf("Sum of %d and %d is: %d\n", *ptr1, *ptr2, sum);
    return 0;
}

```

////////////////////////////////////

//LINEAR SEARCH

`#include <stdio.h>`

`int main()`

`{`

`int arr[50];`

`int size, target;`

// Input the size of the array

`printf("Enter the size of the array: ");`

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

// Input elements into the array

`printf("Enter %d integers: ", size);`

`for (int i = 0; i < size; i++)`

`{`

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

`}`

// Input the element to search

`printf("Enter the element to search: ");`

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

// Perform linear search

`int index = -1; // Assume the element is not found initially`

`for (int i = 0; i < size; i++)`

`{`

`if (arr[i] == target)`

`{`

`index = i; // Update index if the element is found`

`break;`

`}`

`}`

// Display the result

`if (index != -1)`

`{`

`printf("%d found at index %d.\n", target, index);`

`}`

`else`

`{`

`printf("%d not found in the array.\n", target);`

`}`

`return 0;`

`}`

```
////////////////////////////////////  
  
//PRIME NUMBERS UPT A LIMIT  
#include <stdio.h>  
int main()  
{  
    int limit;  
  
    // Input the limit  
    printf("Enter the limit for prime numbers: ");  
    scanf("%d", &limit);  
  
    if (limit <= 1)  
    {  
        printf("Prime numbers start from 2. Enter a limit greater than 1.\n");  
        return 1; // Exit the program with an error code  
    }  
  
    // Display prime numbers up to the limit  
    printf("Prime numbers up to %d are: ", limit);  
  
    for (int num = 2; num <= limit; num++)  
    {  
        int isPrime = 1; // Assume the number is prime initially  
  
        // Check for factors  
        for (int i = 2; i <= num / 2; i++)  
        {  
            if (num % i == 0)  
            {  
                isPrime = 0; // Set isPrime to 0 if the number has a factor  
                break;  
            }  
        }  
  
        // Display the prime number  
        if (isPrime)  
        {  
            printf("%d ", num);  
        }  
    }  
    return 0;  
}
```

```

//READ AND WRITE OPERATIONS ON FILE
#include <stdio.h>
#define MAX_EMPLOYEES 3
#define MAX_NAME_LENGTH 50

int main()
{
    FILE *file;
    char names[MAX_EMPLOYEES][MAX_NAME_LENGTH];
    int employeeIds[MAX_EMPLOYEES];
    float salaries[MAX_EMPLOYEES];

    // Create a file for writing
    file = fopen("employee.txt", "w");

    if (file == NULL)
    {
        printf("Error opening file for writing.\n");
        return 1; // Exit with an error code
    }

    // Write employee details to the file
    printf("Enter employee details:\n");
    for (int i = 0; i < MAX_EMPLOYEES; i++)
    {
        printf("Employee %d\n", i + 1);
        printf("Name: ");
        scanf("%s", names[i]);
        printf("Employee ID: ");
        scanf("%d", &employeeIds[i]);
        printf("Salary: ");
        scanf("%f", &salaries[i]);
        printf("\n");
    }

    // Write employee details to the file
    for (int i = 0; i < MAX_EMPLOYEES; i++)
    {
        fprintf(file, "%s %d %.2f\n", names[i], employeeIds[i], salaries[i]);
    }

    // Close the file
    fclose(file);

    // Open the file for reading

```

```

file = fopen("employee.txt", "r");

if (file == NULL)
{
    printf("Error opening file for reading.\n");
    return 1; // Exit with an error code
}

// Read and display employee details from the file
printf("\nEmployee details from the file:\n");
for (int i = 0; i < MAX_EMPLOYEES; i++)
{
    fscanf(file, "%s %d %f", names[i], &employeeIds[i], &salaries[i]);
    printf("Employee %d\n", i + 1);
    printf("Name: %s\n", names[i]);
    printf("Employee ID: %d\n", employeeIds[i]);
    printf("Salary: %.2f\n", salaries[i]);
    printf("\n");
}

// Close the file
fclose(file);
return 0; // Exit successfully
}

////////////////////////////////////

//SEPERATE NUMBERS TO ODD AND EVEN FILES
#include <stdio.h>
int main()
{
    FILE *inputFile, *evenFile, *oddFile;
    int n, num;

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

    // Open the file for writing
    inputFile = fopen("numbers.txt", "w");

    if (inputFile == NULL)
    {
        printf("Error opening file for writing.\n");
        return 1; // Exit with an error code
    }

```

```

// Input numbers and write them to the file
printf("Enter %d numbers:\n", n);
for (int i = 0; i < n; i++)
{
    scanf("%d", &num);
    fprintf(inputFile, "%d\n", num);
}

// Close the input file
fclose(inputFile);

// Open the input file for reading
inputFile = fopen("numbers.txt", "r");

if (inputFile == NULL)
{
    printf("Error opening file for reading.\n");
    return 1; // Exit with an error code
}

// Open files for even and odd numbers
evenFile = fopen("even_numbers.txt", "w");
oddFile = fopen("odd_numbers.txt", "w");

if (evenFile == NULL || oddFile == NULL)
{
    printf("Error opening files for writing even and odd numbers.\n");
    return 1; // Exit with an error code
}

// Read numbers from the input file and separate into even and odd files
while (fscanf(inputFile, "%d", &num) == 1)
{
    if (num % 2 == 0)
    {
        fprintf(evenFile, "%d\n", num);
    }
    else
    {
        fprintf(oddFile, "%d\n", num);
    }
}

// Close all files
fclose(inputFile);

```

```

fclose(evenFile);
fclose(oddFile);

printf("Even and odd numbers separated into different files.\n");

return 0; // Exit successfully
}

////////////////////////////////////

//COPY CONTENTS OF ONE FILE TO ANOTHER
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *sourceFile, *destinationFile;
    char ch;

    // Check if the correct number of command line arguments is provided
    if (argc != 3)
    {
        printf("Invalid number of arguments. Please provide source and destination\n");
        return 1; // Exit with an error code
    }

    // Open the source file for reading
    sourceFile = fopen(argv[1], "r");
    if (sourceFile == NULL)
    {
        printf("Error opening source file.\n");
        return 1; // Exit with an error code
    }

    // Open the destination file for writing
    destinationFile = fopen(argv[2], "w");
    if (destinationFile == NULL)
    {
        printf("Error opening destination file.\n");
        fclose(sourceFile);
        return 1; // Exit with an error code
    }

    // Copy contents from source file to destination file
    while ((ch = fgetc(sourceFile)) != EOF)
    {
        fputc(ch, destinationFile);
    }
}

```

```

        // Close both files
        fclose(sourceFile);
        fclose(destinationFile);

        printf("File copy successful.\n");

        return 0; // Exit successfully
    }

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//AVERAGE OF NUMBERS IN AN ARRAY
#include <stdio.h>
int main()
{
    int arr[50];
    int n, sum = 0;
    float avg;

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

    // Input elements into the array
    printf("Enter %d integers: ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
        sum = sum + arr[i];
    }

    // Calculate and display the average
    printf("Sum: %d", sum);

    avg = sum/n;
    printf("\nAverage: %f", avg);
    return 0;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//NO OF OCCURENCES OF A DIGIT IN A NUMBER
#include <stdio.h>
int main()
{

```



```

long long number;
int digit;

// Input the number
printf("Enter a number: ");
scanf("%lld", &number);

// Input the digit to find
printf("Enter the digit to find: ");
scanf("%d", &digit);

// Count the occurrences of the digit
int count = 0;
long long tempNumber = number;

while (tempNumber != 0)
{
    int lastDigit = tempNumber % 10;
    if (lastDigit == digit)
    {
        count++;
    }
    tempNumber /= 10;
}

// Display the result
printf("The digit %d occurs %d times in the number %lld.\n", digit, count, number);
return 0;
}

////////////////////////////////////

//STRING OPERATIONS
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[100], str2[100], result[200];

    // Input the first string
    printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    str1[strcspn(str1, "\n")] = '\0'; // Remove the newline character from fgets input

    // Input the second string
    printf("Enter the second string: ");

```

```

fgets(str2, sizeof(str2), stdin);
str2[strcspn(str2, "\n")] = '\0'; // Remove the newline character from fgets i

// Find and display the length of the first string
printf("Length of the first string: %zu\n", strlen(str1));

// Concatenate the two strings using strcat function and display the result
strcpy(result, str1); // Copy the first string to the result
strcat(result, str2); // Concatenate the second string to the result
printf("Concatenated string: %s\n", result);

// Compare the two strings using strcmp function and display the result
int comparisonResult = strcmp(str1, str2);
if (comparisonResult == 0)
{
    printf("The strings are equal.\n");
}
else if (comparisonResult < 0)
{
    printf("The first string is less than the second string.\n");
} else
{
    printf("The first string is greater than the second string.\n");
}

// Reverse the first string using strrev function and display the result
strcpy(result, str1); // Copy the first string to the result
strrev(result); // Reverse the result
printf("Reversed first string: %s\n", result);

return 0;
}

```