

1. C "Hello, World!" Program.

```
#include <stdio.h>
int main()
{
    printf("Hello, World!");
    return 0;
}
```

Output

```
Hello, World!
```

2. C Program to Add Two Integers.

```
#include <stdio.h>
int main()
{
    int number1, number2, sum;

    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    // calculating sum
    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
```

Output

```
Enter two integers: 12
11
12 + 11 = 23
```

3. C Program to Multiply two Floating Point Numbers.

```
#include <stdio.h>
int main()
{
    double a, b, product;
    printf("Enter two numbers: ");
    scanf("%lf %lf", &a, &b);
    product = a * b;
    printf("Product = %.2lf", product);
    return 0;
}
```

Output

```
Enter two numbers: 2.4
1.12
Product = 2.69
```

4. C Program to Find ASCII Value of a Character.

```
#include <stdio.h>
int main()
{
    char c;
    printf("Enter a character: ");
}
```

```

scanf("%c", &c);
printf("ASCII value of %c = %d", c, c);
return 0;
}

```

Output

```

Enter a character: G
ASCII value of G = 71

```

5. C Program to Find the Size of int, float, double and char.

```

#include<stdio.h>
int main()
{
    int intType;
    float floatType;
    double doubleType;
    char charType;
    printf("Size of int: %zu bytes\n", sizeof(intType));
    printf("Size of float: %zu bytes\n", sizeof(floatType));
    printf("Size of double: %zu bytes\n", sizeof(doubleType));
    printf("Size of char: %zu byte\n", sizeof(charType));
    return 0;
}

```

Output

```

Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte

```

6. C Program to Swap Two Numbers using temporary variable.

```

#include<stdio.h>
int main()
{
    double first, second, temp;
    printf("Enter first number: ");
    scanf("%lf", &first);
    printf("Enter second number: ");
    scanf("%lf", &second);
    temp = first;
    first = second;
    second = temp;
    printf("\nAfter swapping, first number = %.2lf\n", first);
    printf("After swapping, second number = %.2lf", second);
    return 0;
}

```

Output

```

Enter first number: 1.20
Enter second number: 2.45

After swapping, first number = 2.45
After swapping, second number = 1.20

```

7. C Program to Swap Two Numbers without using temporary variable.

```
#include<stdio.h>
int main()
{
    double first, second, temp;
    printf("Enter first number: ");
    scanf("%lf", &first);
    printf("Enter second number: ");
    scanf("%lf", &second);
    temp = first;
    first = second;
    second = temp;
    printf("\nAfter swapping, first number = %.2lf\n", first);
    printf("After swapping, second number = %.2lf", second);
    return 0;
}
```

Output

```
Enter first number: 1.20
Enter second number: 2.45

After swapping, first number = 2.45
After swapping, second number = 1.20
```

8. C Program to Check Whether a Number is Even or Odd.

```
#include <stdio.h>
int main()
{
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    if(num % 2 == 0)
        printf("%d is even.", num);
    else
        printf("%d is odd.", num);
    return 0;
}
```

Output

```
Enter an integer: -7
-7 is odd.
```

9. C Program to Check Whether a Character is Vowel or Consonant.

```
#include <stdio.h>
int main() {
    char c;
    int lowercase_vowel, uppercase_vowel;
    printf("Enter an alphabet: ");
    scanf("%c", &c);
    lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

    uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    if (lowercase_vowel || uppercase_vowel)
        printf("%c is a vowel.", c);
    else
        printf("%c is a consonant.", c);
    return 0;
}
```

Output

```
Enter an alphabet: G
G is a consonant.
```

10. C Program to Find the Largest Number Among Three Numbers.

```
#include <stdio.h>
int main() {
    double n1, n2, n3;
    printf("Enter three numbers: ");
    scanf("%lf %lf %lf", &n1, &n2, &n3);
    if (n1 >= n2) {
        if (n1 >= n3)
            printf("%.2lf is the largest number.", n1);
        else
            printf("%.2lf is the largest number.", n3);
    } else {
        if (n2 >= n3)
            printf("%.2lf is the largest number.", n2);
        else
            printf("%.2lf is the largest number.", n3);
    }
    return 0;
}
```

Output

```
Enter three numbers: -4.5
3.9
5.6
5.60 is the largest number.
```

11. C Program to Find all Roots of a Quadratic Equation.

```
#include <math.h>
#include <stdio.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;
    printf("Enter coefficients a, b and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);
    discriminant = b * b - 4 * a * c;
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", root1, root2);
    }
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", root1);
    }
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart, imagPart,
realPart, imagPart);
    }

    return 0;
}
```

Output

```
Enter coefficients a, b and c: 2.3
4
5.6
root1 = -0.87+1.30i and root2 = -0.87-1.30i
```

12. C Program to Check Leap Year.

```
#include <stdio.h>
int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if (year % 400 == 0) {
        printf("%d is a leap year.", year);
    }
    else if (year % 100 == 0) {
        printf("%d is not a leap year.", year);
    }
    else if (year % 4 == 0) {
        printf("%d is a leap year.", year);
    }
    else {
        printf("%d is not a leap year.", year);
    }
    return 0;
}
```

Output 1

```
Enter a year: 1900
1900 is not a leap year.
```

Output 2

```
Enter a year: 2012
2012 is a leap year.
```

13. C Program to Check Whether a Number is Positive or Negative.

```
#include <stdio.h>
int main() {
    double num;
    printf("Enter a number: ");
    scanf("%lf", &num);

    if (num < 0.0)
        printf("You entered a negative number.");
    else if (num > 0.0)
        printf("You entered a positive number.");
    else
        printf("You entered 0.");

    return 0;
}
```

Output

```
Enter a number: 12.3
You entered a positive number.
```

14. C Program to Check Whether a Character is an Alphabet or not.

```
#include <stdio.h>
int main() {
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);
    if (isalpha(c))
        printf("%c is an alphabet.", c);
    else
        printf("%c is not an alphabet.", c);
    return 0;
}
```

Output

```
Enter a character: *
* is not an alphabet
```

15. C Program to Calculate the Sum of Natural Numbers.

```
#include <stdio.h>
int main() {
    int n, i, sum = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    i = 1;
    while (i <= n) {
        sum += i;
        ++i;
    }
    printf("Sum = %d", sum);
    return 0;
}
```

Output

```
Enter a positive integer: 100
Sum = 5050
```

16. C Program to Find Factorial of a Number.

```
#include <stdio.h>
int main() {
    int n, i;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
        }
        printf("Factorial of %d = %llu", n, fact);
    }
    return 0;
}
```

Output

```
Enter an integer: 10
Factorial of 10 = 3628800
```

17. C Program to Generate Multiplication Table.

```
#include <stdio.h>
int main() {
    int n, i;
    printf("Enter an integer: ");
    scanf("%d", &n);
    for (i = 1; i <= 10; ++i) {
        printf("%d * %d = %d \n", n, i, n * i);
    }
    return 0;
}
```

Output

```
Enter an integer: 9
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
```

18. C Program to Display Fibonacci Sequence.

```
#include <stdio.h>
int main() {
    int t1 = 0, t2 = 1, nextTerm = 0, n;
    printf("Enter a positive number: ");
    scanf("%d", &n);
    printf("Fibonacci Series: %d, %d, ", t1, t2);
    nextTerm = t1 + t2;
    while (nextTerm <= n) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }
    return 0;
}
```

Output

```
Enter the number of terms: 10
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

19. C Program to Find GCD of two Numbers.

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

```
int main()
{
    int n1, n2;

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

    while(n1!=n2)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
    printf("GCD = %d", n1);

    return 0;
}
```

Output

```
Enter two positive integers: 81
153
GCD = 9
```

20. C Program to Find LCM of two Numbers.

```
#include <stdio.h>
int main() {
    int n1, n2, max;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    max = (n1 > n2) ? n1 : n2;
    while (1) {
        if (max % n1 == 0 && max % n2 == 0) {
            printf("The LCM of %d and %d is %d.", n1, n2, max);
            break;
        }
        ++max;
    }
    return 0;
}
```

Output

```
Enter two positive integers: 72
120
The LCM of 72 and 120 is 360.
```

21. C Program to Display Characters from A to Z Using Loop.

```
#include <stdio.h>
int main() {
    char c;
    for (c = 'A'; c <= 'Z'; ++c)
        printf("%c ", c);
    return 0;
}
```

Output

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```


22. C Program to Count Number of Digits in an Integer.

```
#include <stdio.h>
int main()
{
    long long n;
    int count = 0;
    printf("Enter an integer: ");
    scanf("%lld", &n);
    do {
        n /= 10;
        ++count;
    } while (n != 0);

    printf("Number of digits: %d", count);
}
```

Output

```
Enter an integer: 3452
Number of digits: 4
```

23. C Program to Reverse a Number.

```
#include <stdio.h>
int main() {
    int n, rev = 0, remainder;
    printf("Enter an integer: ");
    scanf("%d", &n);
    while (n != 0) {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;
    }
    printf("Reversed number = %d", rev);
    return 0;
}
```

Output

```
Enter an integer: 2345
Reversed number = 5432
```

24. C Program to Calculate the Power of a Number.

```
#include <stdio.h>
int main() {
    int base, exp;
    long double result = 1.0;
    printf("Enter a base number: ");
    scanf("%d", &base);
    printf("Enter an exponent: ");
    scanf("%d", &exp);

    while (exp != 0) {
        result *= base;
        --exp;
    }
    printf("Answer = %.0Lf", result);
    return 0;
}
```

Output

```
Enter a base number: 3
Enter an exponent: 4
Answer = 81
```

25. C Program to Check Whether a Number is Palindrome or Not.

```
#include <stdio.h>
int main() {
    int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;
    while (n != 0) {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
        n /= 10;
    }
    if (original == reversed)
        printf("%d is a palindrome.", original);
    else
        printf("%d is not a palindrome.", original);

    return 0;
}
```

Output

```
Enter an integer: 1001
1001 is a palindrome.
```

26. C Program to Check Whether a Number is Prime or Not.

```
#include <stdio.h>
int main() {
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for (i = 2; i <= n / 2; ++i) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }
    if (n == 1) {
        printf("1 is neither prime nor composite.");
    }
    else {
        if (flag == 0)
            printf("%d is a prime number.", n);
        else
            printf("%d is not a prime number.", n);
    }
    return 0;
}
```

Output

```
Enter a positive integer: 29
29 is a prime number.
```

27. C Program to Display Prime Numbers Between Two Intervals.

```
#include <stdio.h>
int main()
{
    int low, high, i, flag;
    printf("Enter two numbers(intervals): ");
    scanf("%d %d", &low, &high);
    printf("Prime numbers between %d and %d are: ", low, high);
    while (low < high) {
        flag = 0;
        if (low <= 1) {
            ++low;
            continue;
        }
        for (i = 2; i <= low / 2; ++i) {
            if (low % i == 0) {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            printf("%d ", low);
        ++low;
    }
    return 0;
}
```

Output

```
Enter two numbers(intervals): 20
50
Prime numbers between 20 and 50 are: 23 29 31 37 41 43 47
```

28. C Program to Check Armstrong Number.

```
#include <stdio.h>
int main() {
    int num, originalNum, remainder, result = 0;
    printf("Enter a three-digit integer: ");
    scanf("%d", &num);
    originalNum = num;
    while (originalNum != 0) {
        remainder = originalNum % 10;
        result += remainder * remainder * remainder;
        originalNum /= 10;
    }
    if (result == num)
        printf("%d is an Armstrong number.", num);
    else
        printf("%d is not an Armstrong number.", num);

    return 0;
}
```

Output

```
Enter a three-digit integer: 371
371 is an Armstrong number.
```

29. C Program to Display Factors of a Number.

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

```

int main() {
    int num, i;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Factors of %d are: ", num);
    for (i = 1; i <= num; ++i) {
        if (num % i == 0) {
            printf("%d ", i);
        }
    }
    return 0;
}

```

Output

```

Enter a positive integer: 60
Factors of 60 are: 1 2 3 4 5 6 10 12 15 20 30 60

```

30. C Program to Make a Simple Calculator Using switch...case.

```

#include <stdio.h>
int main() {
    char op;
    double first, second;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &op);
    printf("Enter two operands: ");
    scanf("%lf %lf", &first, &second);
    switch (op) {
        case '+':
            printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
            break;
        case '-':
            printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
            break;
        case '*':
            printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
            break;
        case '/':
            printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
            break;
        default:
            printf("Error! operator is not correct");
    }
    return 0;
}

```

Output

```

Enter an operator (+, -, *, /): *
Enter two operands: 1.5
4.5
1.5 * 4.5 = 6.8

```

31. C Program to Display Prime Numbers Between Intervals Using Function.

```

#include <stdio.h>
int checkPrimeNumber(int n);
int main() {
    int n1, n2, i, flag;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("Prime numbers between %d and %d are: ", n1, n2);
}

```

```

    for (i = n1 + 1; i < n2; ++i) {
        flag = checkPrimeNumber(i);
        if (flag == 1)
            printf("%d ", i);
    }
    return 0;
}
int checkPrimeNumber(int n) {
    int j, flag = 1;
    for (j = 2; j <= n / 2; ++j) {
        if (n % j == 0) {
            flag = 0;
            break;
        }
    }
    return flag;
}

```

Output

```

Enter two positive integers: 12
30
Prime numbers between 12 and 30 are: 13 17 19 23 29

```

32. C Program to Find the Sum of Natural Numbers using Recursion.

```

#include <stdio.h>
int addNumbers(int n);
int main() {
    int num;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Sum = %d", addNumbers(num));
    return 0;
}
int addNumbers(int n) {
    if (n != 0)
        return n + addNumbers(n - 1);
    else
        return n;
}

```

Output

```

Enter a positive integer: 20
Sum = 210

```

33. C Program to Find Factorial of a Number Using Recursion.

```

#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}
long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}

```

Output

```
Enter a positive integer: 6
Factorial of 6 = 720
```

34. C Program to Convert Binary Number to Decimal and vice-versa.

```
#include <stdio.h>
#include <math.h>
int convert(long long);
int main() {
    long long n;
    printf("Enter a binary number: ");
    scanf("%lld", &n);
    printf("%lld in binary = %d in decimal", n, convert(n));
    return 0;
}
int convert(long long n) {
    int dec = 0, i = 0, rem;
    while (n!=0) {
        rem = n % 10;
        n /= 10;
        dec += rem * pow(2, i);
        ++i;
    }
    return dec;
}
```

Output

```
Enter a binary number: 1101
1101 in binary = 13 in decimal
```

35. C Program to Convert Octal Number to Decimal and vice-versa.

```
#include <stdio.h>
#include <math.h>
int convertDecimalToOctal(int decimalNumber);
int main()
{
    int decimalNumber;
    printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);
    printf("%d in decimal = %d in octal", decimalNumber,
    convertDecimalToOctal(decimalNumber));
    return 0;
}
int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i = 1;
    while (decimalNumber != 0)
    {
        octalNumber += (decimalNumber % 8) * i;
        decimalNumber /= 8;
        i *= 10;
    }
    return octalNumber;
}
```

Output

```
Enter a decimal number: 78
78 in decimal = 116 in octal
```

36. C program to calculate the power using recursion

```
#include <stdio.h>
int power(int n1, int n2);
int main() {
    int base, a, result;
    printf("Enter base number: ");
    scanf("%d", &base);
    printf("Enter power number(positive integer): ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d^%d = %d", base, a, result);
    return 0;
}
int power(int base, int a) {
    if (a != 0)
        return (base * power(base, a - 1));
    else
        return 1;
}
```

Output

```
Enter base number: 3
Enter power number(positive integer): 4
3^4 = 81
```

37. C Program to Find Largest Element of an Array.

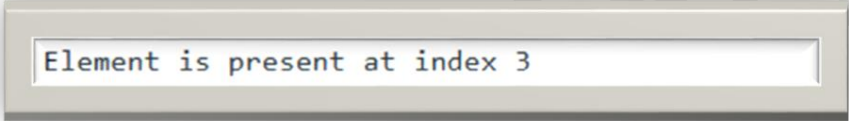
```
#include <stdio.h>
int main() {
    int n;
    double arr[100];
    printf("Enter the number of elements (1 to 100): ");
    scanf("%d", &n);
    for (int i = 0; i < n; ++i) {
        printf("Enter number%d: ", i + 1);
        scanf("%lf", &arr[i]);
    }
    for (int i = 1; i < n; ++i) {
        if (arr[0] < arr[i]) {
            arr[0] = arr[i];
        }
    }
    printf("Largest element = %.2lf", arr[0]);
    return 0;
}
```

Output

```
Enter the number of elements (1 to 100): 5
Enter number1: 34.5
Enter number2: 2.4
Enter number3: -35.5
Enter number4: 38.7
Enter number5: 24.5
Largest element = 38.70
```

38. Write a program to search the given element by using linear search.

```
#include <stdio.h>
int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}
int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = search(arr, n, x);
    (result == -1)
        ? printf("Element is not present in array")
        : printf("Element is present at index %d", result);
    return 0;
}
```

Output


```
Element is present at index 3
```

39. C Program to Add Two Matrix Using Multi-dimensional Arrays.

```
#include <stdio.h>
int main() {
    int r, c, a[10][10], b[10][10], sum[10][10], i, j;
    printf("Enter the number of rows (between 1 and 100): ");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and 10): ");
    scanf("%d", &c);
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }

    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element b%d%d: ", i + 1, j + 1);
            scanf("%d", &b[i][j]);
        }

    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            sum[i][j] = a[i][j] + b[i][j];
        }

    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("%d\t", sum[i][j]);
            if (j == c - 1) {
                printf("\n\n");
            }
        }
}
```



```

    }
    return 0;
}

```

Output

```

Enter the number of rows (between 1 and 100): 2
Enter the number of columns (between 1 and 100): 3

Enter elements of 1st matrix:
Enter element a11: 2
Enter element a12: 3
Enter element a13: 4
Enter element a21: 5
Enter element a22: 2
Enter element a23: 3
Enter elements of 2nd matrix:
Enter element b11: -4
Enter element b12: 5
Enter element b13: 3
Enter element b21: 5
Enter element b22: 6
Enter element b23: 3

Sum of two matrices:
-2  8  7

```

40. C Program to Find Transpose of a Matrix.

```

#include <stdio.h>
int main() {
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    printf("\nEnter matrix elements:\n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    printf("\nEnter matrix: \n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j) {
            printf("%d ", a[i][j]);
            if (j == c - 1)
                printf("\n");
        }
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j) {
            transpose[j][i] = a[i][j];
        }
    printf("\nTranspose of the matrix:\n");
    for (int i = 0; i < c; ++i)
        for (int j = 0; j < r; ++j) {
            printf("%d ", transpose[i][j]);
            if (j == r - 1)
                printf("\n");
        }
}

```

Name :- Divyansh Jaiswal
 Subject :- Computer Science
 return 0;
 }

Roll no. 21UFIA1164

Section:- B4

Output

```
Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7

Entered matrix:
1  4  0
-5  2  7

Transpose of the matrix:
1  -5
4  2
0  7
```

41. C Program to Access Elements of an Array Using Pointer.

```
#include <stdio.h>
int main() {
    int data[5];
    printf("Enter elements: ");
    for (int i = 0; i < 5; ++i)
        scanf("%d", data + i);
    printf("You entered: \n");
    for (int i = 0; i < 5; ++i)
        printf("%d\n", *(data + i));
    return 0;
}
```

Output

```
Enter elements: 1
2
3
5
4
You entered:
1
2
3
5
4
```

42. C Program to Find the Frequency of Characters in a String.

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

```
int main() {
    char str[1000], ch;
    int count = 0;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Enter a character to find its frequency: ");
    scanf("%c", &ch);
    for (int i = 0; str[i] != '\0'; ++i) {
        if (ch == str[i])
            ++count;
    }
    printf("Frequency of %c = %d", ch, count);
    return 0;
}
```

Output

```
Enter a string: This website is awesome.
Enter a character to find its frequency: e
Frequency of e = 4
```

43. C Program to Find the Length of a String.

```
#include <stdio.h>
int main() {
    char s[] = "Programming is fun";
    int i;

    for (i = 0; s[i] != '\0'; ++i);

    printf("Length of the string: %d", i);
    return 0;
}
```

Output

```
Length of the string: 18
```

44. C Program to Concatenate Two Strings.

```
#include <stdio.h>
int main() {
    char s1[100] = "programming ", s2[] = "is awesome";
    int length, j;
    length = 0;
    while (s1[length] != '\0')
    {
        ++length;
    }
    for (j = 0; s2[j] != '\0'; ++j, ++length)
    {
        s1[length] = s2[j];
    }
    s1[length] = '\0';
    printf("After concatenation: ");
    puts(s1);
    return 0;
}
```

Output

After concatenation: programming is awesome

45. C Program to Copy String Without Using strcpy().

```
#include <stdio.h>
int main() {
    char s1[100], s2[100], i;
    printf("Enter string s1: ");
    fgets(s1, sizeof(s1), stdin);
    for (i = 0; s1[i] != '\0'; ++i) {
        s2[i] = s1[i];
    }
    s2[i] = '\0';
    printf("String s2: %s", s2);
    return 0;
}
```

Output

```
Enter string s1: Hey fellow programmer.
String s2: Hey fellow programmer.
```

46. C Program to Store Information of a Student Using Structure.

```
#include <stdio.h>
struct student {
    char name[50];
    int roll;
    float marks;
} s;

int main() {
    printf("Enter information:\n");
    printf("Enter name: ");
    fgets(s.name, sizeof(s.name), stdin);

    printf("Enter roll number: ");
    scanf("%d", &s.roll);
    printf("Enter marks: ");
    scanf("%f", &s.marks);

    printf("Displaying Information:\n");
    printf("Name: ");
    printf("%s", s.name);
    printf("Roll number: %d\n", s.roll);
    printf("Marks: %.1f\n", s.marks);

    return 0;
}
```

Output

```

Enter information:
Enter name: Jack
Enter roll number: 23
Enter marks: 34.5
Displaying Information:
Name: Jack
Roll number: 23
Marks: 34.5

```

47. C Program to Add Two Distances (in inch-feet) System Using Structures.

```

#include <stdio.h>
struct Distance {
    int feet;
    float inch;
} d1, d2, result;
int main() {
    printf("Enter 1st distance\n");
    printf("Enter feet: ");
    scanf("%d", &d1.feet);
    printf("Enter inch: ");
    scanf("%f", &d1.inch);

    printf("\nEnter 2nd distance\n");
    printf("Enter feet: ");
    scanf("%d", &d2.feet);
    printf("Enter inch: ");
    scanf("%f", &d2.inch);

    result.feet = d1.feet + d2.feet;
    result.inch = d1.inch + d2.inch;

    while (result.inch >= 12.0) {
        result.inch = result.inch - 12.0;
        ++result.feet;
    }
    printf("\nSum of distances = %d\'-%.1f\'", result.feet, result.inch);
    return 0;
}

```

Output

```

Enter 1st distance
Enter feet: 23
Enter inch: 8.6

Enter 2nd distance
Enter feet: 34
Enter inch: 2.4

```

48. C Program to Store Information Using Structures with Dynamically Memory Allocation

```

#include <stdio.h>
#include <stdlib.h>
struct course {
    int marks;
    char subject[30];
};

```

```
int main() {
    struct course *ptr;
    int noOfRecords;
    printf("Enter the number of records: ");
    scanf("%d", &noOfRecords);

    ptr = (struct course *)malloc(noOfRecords * sizeof(struct course));
    for (int i = 0; i < noOfRecords; ++i) {
        printf("Enter subject and marks:\n");
        scanf("%s %d", (ptr + i)->subject, &(ptr + i)->marks);
    }
    printf("Displaying Information:\n");
    for (int i = 0; i < noOfRecords; ++i) {
        printf("%s\t%d\n", (ptr + i)->subject, (ptr + i)->marks);
    }
    free(ptr);
    return 0;
}
```

Output

```
Enter the number of records: 2
Enter subject and marks:
Science 82
Enter subject and marks:
DSA 73

Displaying Information:
Science      82
DSA          73
```

49. C Program to Write a Sentence to a File.

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

int main() {
    char sentence[1000];

    // creating file pointer to work with files
    FILE *fptr;

    // opening file in writing mode
    fptr = fopen("program.txt", "w");

    // exiting program
    if (fptr == NULL) {
        printf("Error!");
        exit(1);
    }
    printf("Enter a sentence:\n");
    fgets(sentence, sizeof(sentence), stdin);
    fprintf(fptr, "%s", sentence);
    fclose(fptr);
    return 0;
}
```

Output

Enter a sentence: C Programming is fun

50. C Program to Read a Line From a File and Display it.

```
#include <stdio.h>
#include <stdlib.h> // For exit() function
int main() {
    char c[1000];
    FILE *fptr;
    if ((fptr = fopen("program.txt", "r")) == NULL) {
        printf("Error! File cannot be opened.");
        // Program exits if the file pointer returns NULL.
        exit(1);
    }

    // reads text until newline is encountered
    fscanf(fptr, "%[^\n]", c);
    printf("Data from the file:\n%s", c);
    fclose(fptr);

    return 0;
}
```

Output

```
C programming is awesome.
I love C programming.
How are you doing?

Data from the file:
C programming is awesome.
```

51. C Program to Display its own Source Code as Output.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp = NULL;
    int ch = 0;
    fp = fopen(__FILE__, "r");
    if(fp == NULL)
    {
        printf("Error in opening the file\n");
        exit(1);
    }
    do
    {
        ch = fgetc(fp);
        printf("%c", ch);
    }
    while (ch != EOF);
    fclose (fp);
    return 0;
}
```

Output

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

int main()
{
    //file pointer
    FILE *fp = NULL;
    int ch = 0;
    //open the file
    fp = fopen(__FILE__, "r");
    if(fp == NULL)
    {
        printf("Error in opening the file\n");
        exit(1);
    }
    //read till EOF
    do
    {
        ch = fgetc(fp);
        printf("%c",ch);
    }
    while (ch != EOF);

    //close open file
    fclose (fp);

    return 0;
}

Process returned 0 (0x0)    execution time : 0.107 s
```