



Practical File Of Fundamentals of C Programming(23CS003)

Submitted

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING



**CHANDIGARH-PATIALA NATIONAL HIGHWAY RAJPURA
(PATIALA) PUNJAB-140401 (INDIA)**

Submitted To:

Faculty name: Ms. Kushnian Kaur
Department of Computer Science & Engg.
Chitkara University, Punjab

Submitted By:

Name - Kanav Kumar
Roll No.- 2310992111
Batch/Sem: B.E-CSE/2nd Sem

INDEX

| Sr.No. | Experiment Name | Date | Page No. | Teacher Sign |
|--------|---|----------|----------|--------------|
| 1 | Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h". | 11/01/24 | 1-2 | |
| 2 | Write a program to add two numbers and display the sum. | 18/01/24 | 3 | |
| 3 | Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user. | 19/01/24 | 4 | |
| 4 | Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user. | 24/01/24 | 5 | |
| 5 | Write a program to evaluate each of the following equations. (i) $V = u + at$. (ii) $S = ut + \frac{1}{2}at^2$ (iii) $T = 2a + \sqrt{b+9c}$ (iv) $H = \sqrt{b^2 + p^2}$ | 29/01/24 | 6-9 | |
| 6 | Write a program to swap two variables: a) By using a temporary variable. b) Without using temporary variable | 01/02/24 | 10-11 | |
| 7 | Write a Program to find the greatest among three numbers using: ● Conditional Operator ● If-Else statement | 05/02/24 | 12-14 | |
| 8 | Write the following programs using switch case statement: ● To check that an input alphabet is vowel or consonant ● To check whether a number is positive, negative or zero | 08/02/24 | 15-18 | |

| Sr.No. | Experiment Name | Date | Page No | Teacher Sign |
|--------|--|----------|---------|--------------|
| 9 | Write a program using a while loop to print the sum of first n natural numbers. | 19/02/24 | 19 | |
| 10 | Write a program to check if a number is Armstrong or not using For loop. | 21/02/24 | 20 | |
| 11 | Write the program to count the digits in a number and then print the reverse of the number also. | 22/02/24 | 21 | |
| 12 | Write a program to generate the Fibonacci series. | 28/02/24 | 22 | |
| 13 | <p>Write a program to print the following patterns:</p> <p>a)</p> <pre> * </pre> <p>b)</p> <pre> * * * * * * * * * * * * * * * * * * * * * * * * * </pre> | 01/03/24 | 23 - 26 | |

| Sr. No. | Experiment Name | Date | Page No | Teacher Sign |
|---------|--|----------|---------|--------------|
| 14 | Write the program to print the following pattern: 1 2 3 4 5 6 2 4 6 8 10 12 3 6 9 12 15 18 4 8 12 16 20 24 5 10 15 20 25 30 6 12 18 24 30 36 | 06/03/24 | 27 | |
| 15 | Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions. | 11/03/24 | 28 - 29 | |
| 16 | Write a program to calculate the area and circumference of a circle using functions. | 15/03/24 | 30 - 31 | |
| 17 | Write a program to swap two variables using the concept of call by value and call by reference. | 21/03/24 | 32 | |
| 18 | Write a program to perform the following operations on 1D-Array: <ul style="list-style-type: none"> • Insert • Update • Delete • Display • Linear Search • Binary Search | 28/03/24 | 33 - 43 | |
| 19 | Write a program to calculate the sum of array elements by passing it to a function. | 29/03/24 | 44 - 45 | |
| 20 | Write a program to show the use of passing pointers as arguments to the functions. | 01/04/24 | 46 | |

| | | | | |
|----|---|----------|---------|--|
| 21 | Write a program matrix multiplication using the concept of 2D | 03/04/24 | 47 - 48 | |
| 22 | Write a program to transpose a given matrix. | 04/04/24 | 49 | |
| 23 | Write a program to find the factorial of a number by using the concept of recursion. | 04/04/24 | 50 | |
| 24 | Write a menu driven C program to show the use of in-built string functions like strlen, strcat, strcpy, strcmp, strcmp, strcmp etc. | 17/04/24 | 51-53 | |
| 25 | Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string. | 17/04/24 | 54 | |
| 26 | Write a program to display the sum of the digits of a number by using the concept of recursion. | 22/04/24 | 55 | |
| 27 | Write a C program to add two distances in inch & feet using the concept of structures. | 22/04/24 | 56-57 | |
| 28 | Write a C program to add two complex numbers using the concept of structures in C. | 02/05/24 | 58-59 | |
| 29 | Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure. | 06/05/24 | 60-61 | |
| 30 | Write a Program in C to store and retrieve the information about students of a university by using the concept of file handling. | 08/05/24 | 62-63 | |
| 31 | Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string. | 08/05/24 | 64-65 | |



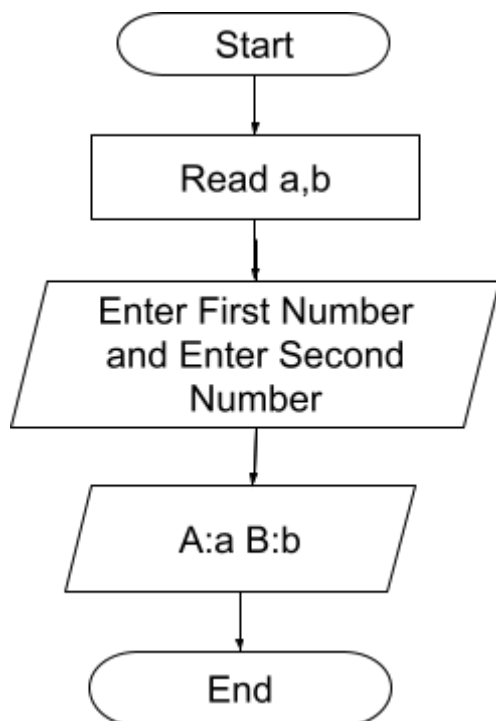
Experiment No. 1

Aim: Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h".

Program Used: Visual Studio Code

Solution:

Flowchart:



Program Code:

```
#include<stdio.h>

int main(){
    int a, b;
    printf("Enter first number : ");
    scanf("%d", &a);
    printf("Enter second number : ");
    scanf("%d", &b);
    printf("A:%d\tB:%d", a, b);
    return 0;
}
```

Output:

```
Enter first number : 10
Enter second number : 5
A:10    B:5
```

Experiment No. 2

Aim: Write a program to add two numbers and display the sum.

Program Code:

```
#include <stdio.h>

int main()
{
    int x, y, sum;
    printf("Enter value of x and y : ");
    scanf("%d %d", &x, &y);
    sum = x + y;
    printf("Sum of x and y is : %d", sum);
    return 0;
}
```

Output:

```
Enter value of x and y : 100 50
Sum of x and y is : 150
```


Experiment No. 3

Aim: Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user.

Program Code:

```
#include <stdio.h>

int main()
{
    float radius, pi, area, circumference;
    printf("Enter value of radius : ");
    scanf("%f", &radius);
    pi = 3.14159;
    area = pi * radius * radius;
    circumference = 2 * pi * radius;
    printf("Area of the given radius is : %f\n", area);
    printf("Circumference of the circle is : %f", circumference);
    return 0;
}
```

Output:

```
Enter value of radius : 5
Area of the given radius is : 78.539749
Circumference of the circle is : 31.415901%
```

Experiment No. 4

Aim: Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user.

Program Code:

```
#include <stdio.h>

int main()
{
    int x, y, sum, minus, multiply;
    float divide;
    printf("Enter values of x and y : ");
    scanf("%d %d", &x, &y);
    sum = x + y;
    minus = x - y;
    multiply = x * y;
    divide = (float)x / y;
    printf("Summation of x and y is: %d\n", sum);
    printf("Subtraction of x and y is: %d\n", minus);
    printf("Multiplication of x and y is: %d\n", multiply);
    printf("Division of x and y is: %.2f\n", divide);
    return 0;
}
```

Output:

```
Enter values of x and y : 100 50
Summation of x and y is: 150
Subtraction of x and y is: 50
Multiplication of x and y is: 5000
Division of x and y is: 2.00
```

Experiment No. 5

Aim: Write a program to evaluate each of the following equations.

(i) $V = u + at$ (ii) $S = ut + \frac{1}{2}at^2$ (iii) $T = 2a + \sqrt{b+9c}$ (iv) $H = \sqrt{b^2+p^2}$

(i) $V = u + at$

Program Code:

```
#include <stdio.h>

int main()
{
    int V, u, a, t;
    printf("Enter the initial velocity (u): ");
    scanf("%d", &u);
    printf("Enter the acceleration (a): ");
    scanf("%d", &a);
    printf("Enter the time (t): ");
    scanf("%d", &t);
    V = u + a * t;
    printf("Final velocity (V) is %d\n", V);
    return 0;
}
```

Output:

```
Enter the initial velocity (u): 10
Enter the acceleration (a): 9
Enter the time (t): 10
Final velocity (V) is 100
```

(ii) $S = ut + \frac{1}{2}at^2$

Program Code:

```
#include <stdio.h>

int main()
{
    int S, u, a, t;
    printf("Enter the initial velocity (u): ");
    scanf("%d", &u);
    printf("Enter the acceleration (a): ");
    scanf("%d", &a);
    printf("Enter the time (t): ");
    scanf("%d", &t);
    S = u * t + 0.5 * a * t * t;
    printf("Displacement (S) is %d\n", S);
    return 0;
}
```

Output:

```
Enter the initial velocity (u): 10
Enter the acceleration (a): 8
Enter the time (t): 5
Displacement (S) is 150
```

(iii) $T = 2a + \sqrt{b} + 9c$

Program Code:

```
#include <stdio.h>
#include <math.h>
int main()
{
    double T, a, b, c;
    printf("Enter the value of 'a': ");
    scanf("%lf", &a);
    printf("Enter the value of 'b': ");
    scanf("%lf", &b);
    printf("Enter the value of 'c': ");
    scanf("%lf", &c);
    T = 2 * a + sqrt(b) + 9 * c;
    printf("The result of the equation  $T = 2a + \sqrt{b} + 9c$  is: %lf\n", T);
    return 0;
}
```

Output:

```
Enter the value of 'a': 9
Enter the value of 'b': 5
Enter the value of 'c': 4
The result of the equation  $T = 2a + \sqrt{b} + 9c$  is: 56.236068
```

(iv) $H = \sqrt{b^2 + p^2}$

Program Code:

```
#include <stdio.h>
#include <math.h>
int main()
{
    double H, b, p;
    printf("Enter the value of 'b': ");
    scanf("%lf", &b);
    printf("Enter the value of 'p': ");
    scanf("%lf", &p);
    H = sqrt(b * b) + p * p;
    printf("The result of the equation H = sqrt(b^2 + p^2) is: %lf\n", H);

    return 0;
}
```

Output:

```
Enter the value of 'b': 10
Enter the value of 'p': 5
The result of the equation H = sqrt(b*2) + p*2 is: 35.000000
```

Experiment No. 6

Aim: Write a program to swap two variables:

- a) By using a temporary variable.
- b) Without using temporary variable

a) By using a temporary variable

Program Code:

```
#include <stdio.h>

int main()
{
    int Temp, x, y;
    printf("Enter values of x and y: ");
    scanf("%d %d", &x, &y);
    Temp = x;
    x = y;
    y = Temp;
    printf("Swapped numbers are: x = %d, y = %d\n", x, y);
    return 0;
}
```

Output:

```
Enter values of x and y: 10 20
Swapped numbers are: x = 20, y = 10
```

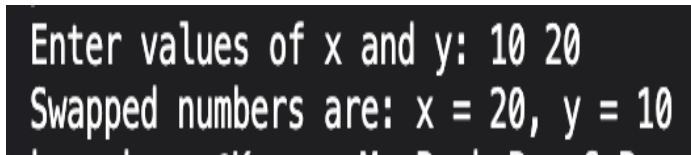
b) Without using temporary variable

Program Code:

```
#include <stdio.h>

int main()
{
    int x, y;
    printf("Enter values of x and y: ");
    scanf("%d %d", &x, &y);
    x = x + y;
    y = x - y;
    x = x - y;
    printf("Swapped numbers are: x = %d, y = %d\n", x, y);
    return 0;
}
```

Output:



```
Enter values of x and y: 10 20
Swapped numbers are: x = 20, y = 10
```


Experiment No. 7

Aim: Write a Program to find the greatest among three numbers using:

- Conditional Operator
- If-Else statement

1) Conditional Operator

Program Code:

```
#include <stdio.h>

int main()
{
    int x, y, z, greatest;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &x, &y, &z);
    greatest = (x > y) ? ((x > z) ? x : z) : ((y > z) ? y : z);
    printf("The greatest number is: %d\n", greatest);
    return 0;
}
```

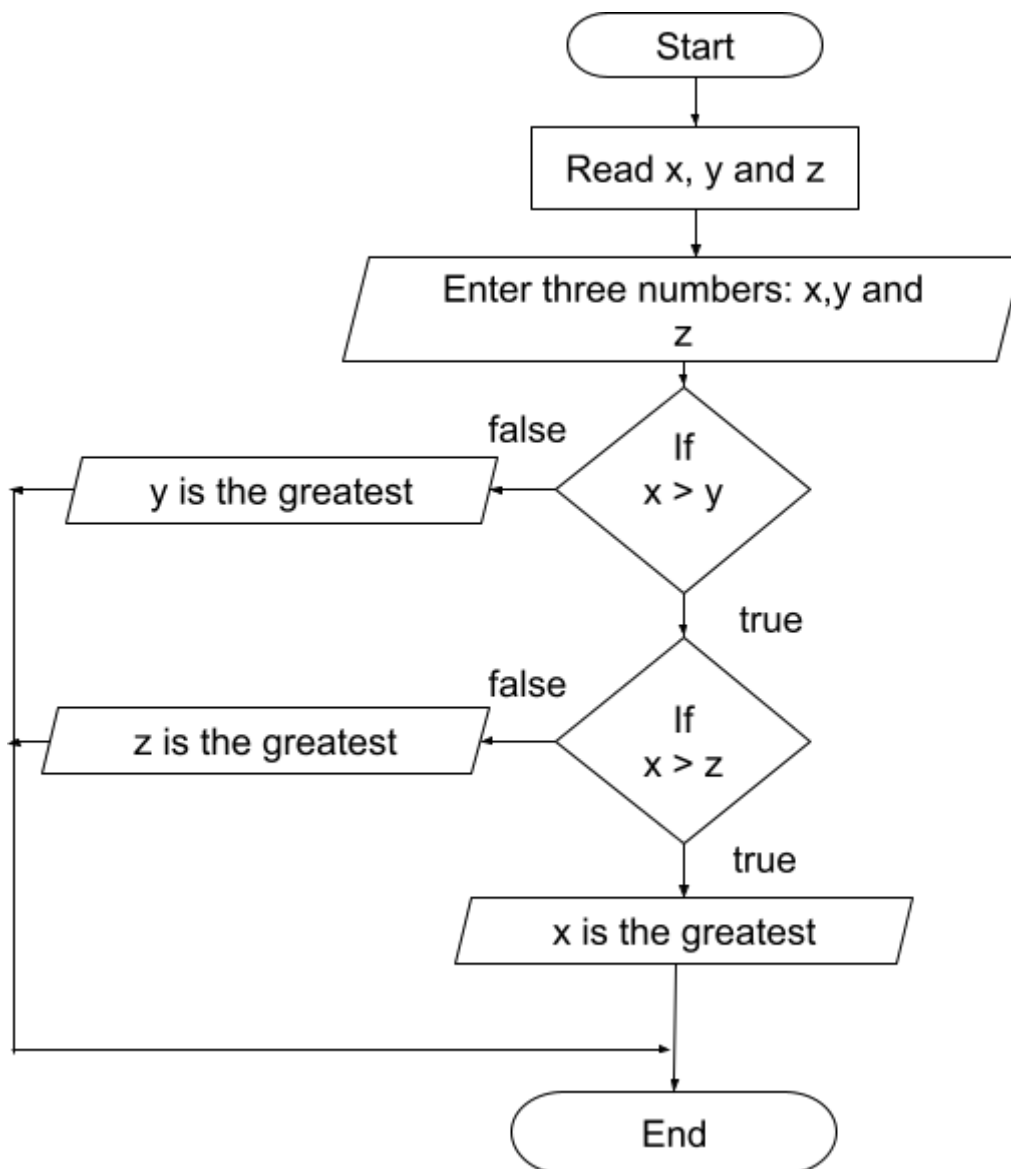
Output:

```
Enter three numbers: 12 13 15
The greatest number is: 15
```

2) If-Else statement

Solution:

Flowchart:



Program Code:

```
#include <stdio.h>

int main() {
    int x, y, z;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &x, &y, &z);
    if (x > y) {
        if (x > z) {
            printf("greatest is : %d", x);
        } else {
            printf("greatest is : %d", z);
        }
    } else {
        if (y > z) {
            printf("greatest is : %d", y);
        } else {
            printf("greatest is : %d", z);
        }
    }
    return 0;
}
```

Output:

```
Enter three numbers: -1 0 1
greatest is : 1
```

Experiment No. 8

Aim: Write the following programs using switch case statement:

- To check that an input alphabet is vowel or consonant
- To check whether a number is positive, negative or zero

1) To check that an input alphabet is vowel or consonant

Program Code:

```
#include <stdio.h>

int main()
{
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);
    switch (c)
    {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            printf("This character '%c' is a vowel\n", c);
            break;
        default:
            printf("This character '%c' is a consonant\n", c);
    }
}
```

```
return 0;  
}
```

Output:

```
Enter a character: e  
This character 'e' is a vowel
```

2) To check whether a number is positive, negative or zero

Program Code:

```
#include <stdio.h>

int main()
{
    int num;
    printf("Enter any number: ");
    scanf("%d", &num);
    switch (num > 0)
    {
        case 1:
            printf("%d is positive.", num);
            break;
        case 0:
            switch (num < 0)
            {
                case 1:
                    printf("%d is negative.", num);
                    break;
                case 0:
                    printf("%d is zero.", num);
                    break;
            }
            break;
    }
}
```



```
}  
return 0;  
}
```

Output:

```
Enter any number: -1  
-1 is negative.%
```

Experiment No. 9

Aim: Write a program using a while loop to print the sum of first n natural numbers.

Program Code:

```
#include <stdio.h>

int main()
{
    int i = 1, num, sum = 0;
    printf("Enter num : ");
    scanf("%d", &num);
    while(i <= num){
        sum += i;
        i++;
    }
    printf("Sum of first %d natural number is : %d", num, sum);

    return 0;
}
```

Output:

```
Enter num : 5
Sum of first 5 natural number is : 15
```


Experiment No. 10

Aim: Write a program to check if a number is Armstrong or not using For loop.

Program Code:

```
#include <stdio.h>

int main() {
    int n, r, arm = 0, i;
    printf("Enter value of n : ");
    scanf("%d", &n);
    for(i = n; i > 0; i = i / 10) {
        r = i % 10;
        arm = (r * r * r) + arm;
    }
    if (n == arm) {
        printf("%d is an Armstrong number\n", n);
    } else {
        printf("%d is not an Armstrong number\n", n);
    }
    return 0;
}
```

Output:

```
Enter value of n : 153
153 is an Armstrong number
```

Experiment No. 11

Aim: Write the program to count the digits in a number and then print the reverse of the number also.

Program Code:

```
#include <stdio.h>

int main() {
    int num, count = 0, reverse = 0, rem;
    printf("Enter value of num: ");
    scanf("%d", &num);
    while(num != 0){
        rem = num % 10;
        reverse = reverse * 10 + rem;
        num = num / 10;
        count++;
    }
    printf("Count is : %d\n", count);
    printf("Reverse of the num is : %d\n", reverse);

    return 0;
}
```

Output:

```
Enter value of num: 123456
Count is : 6
Reverse of the num is : 654321
```

Experiment No. 12

Aim: Write a program to generate the Fibonacci series.

Program Code:

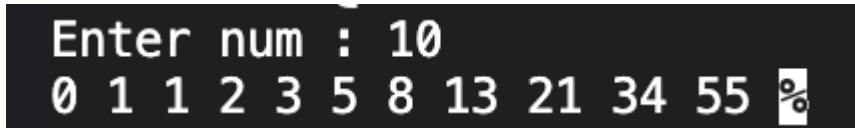
```
#include <stdio.h>

int main() {
    int a = 0, b = 1, c, n;
    printf("Enter num : ");
    scanf("%d", &n);

    for(int i = 0; i <= n; i++){
        printf("%d ", a);
        c = a + b;
        a = b;
        b = c;
    }

    return 0;
}
```

Output:



```
Enter num : 10
0 1 1 2 3 5 8 13 21 34 55 %
```

Experiment No. 13

Aim: Write a program to print the following patterns:

a)

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

Program Code:

a)

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



Output:

```
Enter number : 6
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

b)

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

Program Code:

```
#include <stdio.h>

int main() {
    int i,j,k,n;
    printf("Enter number : ");
    scanf("%d", &n);
    for(i = 1; i <= n; i++){
        for(j = i; j < n; j++){
            printf(" ");
        }
        for(k = 1; k <= i; k++){
            printf(" * ");
        }
        printf("\n");
    }
    return 0;
}
```



Output:

```
Enter number : 6
                *
              * *
            * * *
          * * * *
        * * * * *
      * * * * *
    * * * * *
```

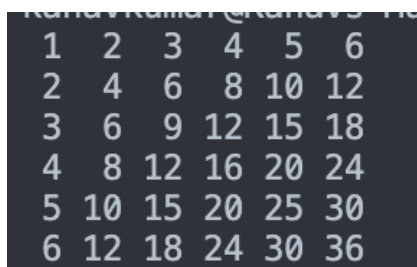
Experiment No. 14

Aim: Write the program to print the following pattern:

```
1 2 3 4 5 6
2 4 6 8 10 12
3 6 9 12 15 18
4 8 12 16 20 24
5 10 15 20 25 30
6 12 18 24 30 36
```

Program Code:

```
int rows = 6, cols = 6;
for(int i = 1; i <= rows; i++) {
    for(int j = 1; j <= cols; j++) {
        printf("%2d ", i * j);
    }
    printf("\n");
}
```

Output:

```
1 2 3 4 5 6
2 4 6 8 10 12
3 6 9 12 15 18
4 8 12 16 20 24
5 10 15 20 25 30
6 12 18 24 30 36
```

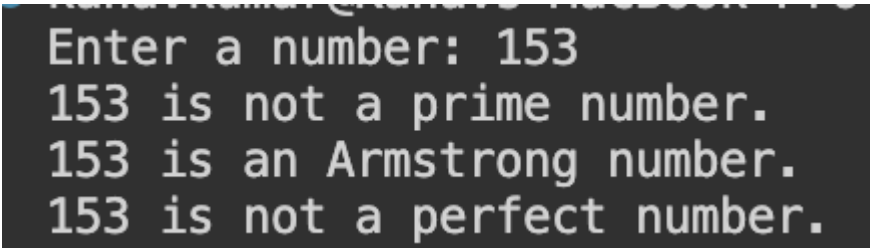

Experiment No. 15

Aim: Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions.

Program Code:

```
#include <stdio.h>
int isPrime(int num) {
    if (num == 1 || num < 0)
        return 0;
    for (int i = 2; i < num; i++) {
        if (num % i == 0)
            return 0;
    }
    return 1;
}
int isArmstrong(int num) {
    int c = num, r, arm = 0;
    while (num > 0) {
        r = num % 10;
        arm = (r * r * r) + arm;
        num = num / 10;
    }
    return c == arm;
}
int isPerfect(int num) {
    int sum = 0;
    for (int i = 1; i < num; i++) {
        if (num % i == 0) {
            sum = sum + i;
        }
    }
    return sum == num;
}
```

```
int main() {  
    int num;  
    printf("Enter a number: ");  
    scanf("%d", &num);  
    if (isPrime(num)) {  
        printf("%d is a prime number.\n", num);  
    } else {  
        printf("%d is not a prime number.\n", num);  
    }  
    if (isArmstrong(num)) {  
        printf("%d is an Armstrong number.\n", num);  
    } else {  
        printf("%d is not an Armstrong number.\n", num);  
    }  
    if (isPerfect(num)) {  
        printf("%d is a perfect number.\n", num);  
    } else {  
        printf("%d is not a perfect number.\n", num);  
    }  
    return 0;  
}
```

Output:

```
Enter a number: 153  
153 is not a prime number.  
153 is an Armstrong number.  
153 is not a perfect number.
```

Experiment No. 16:

Aim: Write a program to calculate the area and circumference of a circle using functions.

Program Code:

```
#include <stdio.h>

float areaCircle(float radius);
float circumferenceCircle(float radius);

int main(){
    float radius;
    printf("Enter value of radius : ");
    scanf("%f", &radius);
    float area = areaCircle(radius);
    float circumference = circumferenceCircle(radius);
    printf("Area of the circle is : %.2f\n", area);
    printf("Circumference of the circle is : %.2f\n", circumference);
    return 0;
}

float areaCircle(float radius){
    return 3.14159 * radius * radius;
}

float circumferenceCircle(float radius){
    return 2 * 3.14159 * radius;
}
```

Output:

```
Enter value of radius : 5  
Area of the circle is : 78.54  
Circumference of the circle is : 31.42
```

Experiment No. 17

Aim: Write a program to swap two variables using the concept of call by value and call by reference.

a) Call by Value:

Program Code:

```
#include <stdio.h>

void swapNum(int a, int b){
    int Temp;
    Temp = a;
    a = b;
    b = Temp;
    printf("After swapping numbers are : a = %d and b = %d\n",a, b);
}

int main(void){
    int a, b;
    printf("Enter 2 numbers : ");
    scanf("%d %d", &a, &b);
    printf("Before swapping numbers are : a = %d and b = %d\n",a, b);
    swapNum(a, b);
}
```

Output:

```
Enter 2 numbers : 5 10
Before swapping numbers are : a = 5 and b = 10
After swapping numbers are : a = 10 and b = 5
```

b) Call by Reference:**Program Code:**

```
#include <stdio.h>

void swapNum(int* a, int* b){
    int Temp;
    Temp = *a;
    *a = *b;
    *b = Temp;
    printf("After swapping numbers are : a = %d and b = %d\n", *a, *b);
}

int main(void){
    int a, b;
    printf("Enter 2 numbers : ");
    scanf("%d %d", &a, &b);
    printf("Before swapping numbers are : a = %d and b = %d\n", a, b);
    swapNum(&a, &b);
}
```

Output:

```
Enter 2 numbers : 100 200
Before swapping numbers are : a = 100 and b = 200
After swapping numbers are : a = 200 and b = 100
```

Experiment No. 18:

Aim: Write a program to perform the following operations on 1D-Array:

- **Insert**
- **Update**
- **Delete**
- **Display**
- **Linear Search**
- **Binary Search**

1) Insert

Program Code:

```
#include <stdio.h>

int main() {
    int array[100], n, x, pos;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    printf("Enter the elements: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &array[i]);
    }
    printf("Enter the new element to be inserted: ");
    scanf("%d", &x);
    printf("Enter the position where the element is to be inserted: ");
    scanf("%d", &pos);
```

```
for (int i = n; i >= pos; i--) {  
    array[i] = array[i - 1];  
}  
array[pos - 1] = x;  
n++;  
printf("Modified array elements are: ");  
for (int i = 0; i < n; i++) {  
    printf("%d ", array[i]);  
}  
return 0;  
}
```

Output:

```
KanavKumar@Kanav's MacBook Pro:CS003 % gcc 7.c -o 7 -lcs003; KanavKumar % ./7  
Enter the number of elements in the array: 5  
Enter the elements: 1 2 3 4 6  
Enter the new element to be inserted: 5  
Enter the position where the element is to be inserted: 5  
Modified array elements are:  
1 2 3 4 5 6 %
```


2) Update

Program Code:

```
#include <stdio.h>

int main() {
    int arr[100];
    int size;
    int ele, pos;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements of the array: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter new element: ");
    scanf("%d", &ele);
    printf("Enter position to update (0-indexed): ");
    scanf("%d", &pos);
    arr[pos] = ele;
    printf("Element updated successfully.\n");
    printf("Updated array:");
    for (int i = 0; i < size; i++) {
        printf(" %d", arr[i]);
    }
    printf("\n");
    return 0;
}
```

Output:

```
Enter size of the array: 5
Enter elements of the array: 1 2 3 3 5
Enter new element: 4
Enter position to update (0-indexed): 3
Element updated successfully.
Updated array: 1 2 3 4 5
```

3) Delete

Program Code:

```
#include <stdio.h>

int main() {
    int arr[100];
    int size;
    int pos;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements of the array: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter position to delete (0-indexed): ");
    scanf("%d", &pos);
    for (int i = pos; i < size - 1; i++) {
        arr[i] = arr[i + 1];
    }
    size--;
    printf("Updated array:");
    for (int i = 0; i < size; i++) {
        printf(" %d", arr[i]);
    }
    printf("\n");
    return 0;
}
```

Output:

```
Enter size of the array: 5
Enter elements of the array: 1 2 3 5 4
Enter position to delete (0-indexed): 3
Updated array: 1 2 3 4
```

4) Display

Program Code:

```
#include <stdio.h>

int main() {
    int arr[100];
    int size;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements of the array: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Array elements:");
    for (int i = 0; i < size; i++) {
        printf(" %d", arr[i]);
    }
    printf("\n");
    return 0;
}
```

Output:

```
Enter size of the array: 5
Enter elements of the array: 9 8 7 6 5
Array elements: 9 8 7 6 5
```

5) Linear Search

Program Code:

```
#include <stdio.h>

int main() {
    int array[100], size, element, found = 0;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    printf("Enter elements of array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &element);
    for (int i = 0; i < size; i++) {
        if (array[i] == element) {
            printf("Element found at index %d\n", i);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Element not found in the array.\n");
    }
    return 0;
}
```

Output:

```
Enter the size of the array: 5
Enter elements of array:
5 10 15 20 25
Enter the element to search: 10
Element found at index 1
```

6) Binary Search

Program Code:

```
#include <stdio.h>

int binarySearch(int arr[], int size, int key){
    int low = 0;
    int high = size - 1;
    while (low <= high){
        int mid = low + (high - low) / 2;
        if (arr[mid] == key){
            return mid;
        } if (arr[mid] < key){
            low = mid + 1;
        } else{
            high = mid - 1;
        }
    }
    return -1;
}

int main(){
    int size, key;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter %d elements in ascending order:\n", size);
    for (int i = 0; i < size; i++){
        scanf("%d", &arr[i]);
    }
}
```



```
}  
printf("Enter key to search: ");  
scanf("%d", &key);  
int result = binarySearch(arr, size, key);  
if (result != -1){  
    printf("Element found at index %d.\n", result);  
}else{  
    printf("Element not found in the array.\n");  
}  
return 0;  
}
```

Output:

```
Enter size of the array: 5  
Enter 5 elements in ascending order:  
1 3 5 7 9  
Enter key to search: 7  
Element found at index 3.
```

Experiment No. 19

Aim: Write a program to calculate the sum of array elements by passing it to a function.

Program Code:

```
#include <stdio.h>

int sumOfArray(int arr[], int size);

int main() {
    int arr[100];
    int size;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements of the array: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    int sum = sumOfArray(arr, size);
    printf("Sum of array elements: %d\n", sum);
    return 0;
}

int sumOfArray(int arr[], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    return sum;
}
```

Output:

```
Enter size of the array: 5  
Enter elements of the array: 1 3 5 7 9  
Sum of array elements: 25
```

Experiment No. 20

Aim: Write a program to show the use of passing pointers as arguments to the functions.

Program Code:

```
#include <stdio.h>

void swap(int *a, int *b);

int main() {
    int x = 10, y = 20;
    printf("Before swapping: x = %d, y = %d\n", x, y);
    swap(&x, &y);
    printf("After swapping: x = %d, y = %d\n", x, y);

    return 0;
}

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Output:

```
Before swapping: x = 10, y = 20
After swapping: x = 20, y = 10
```

Experiment No. 21

Aim: Write a program matrix multiplication using the concept of 2D

Program Code:

```
#include <stdio.h>

int main() {
    int a[3][3], b[3][3], mul[3][3];
    int i, j, k;
    printf("Enter the elements of the first matrix (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter the elements of the second matrix (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            scanf("%d", &b[i][j]);
        }
    }
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            mul[i][j] = 0;
            for (k = 0; k < 3; k++) {
                mul[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```

```
    }  
}  
printf("Multiplication of the matrices:\n");  
for (i = 0; i < 3; i++) {  
    for (j = 0; j < 3; j++) {  
        printf("%d ", mul[i][j]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

Output:

```
Enter the elements of the first matrix (3x3):  
1 2 3  
4 5 6  
7 8 9  
Enter the elements of the second matrix (3x3):  
2 2 2  
2 2 2  
2 2 2  
Multiplication of the matrices:  
12 12 12  
30 30 30  
48 48 48
```

Experiment No. 22

Aim: Write a program to transpose a given matrix.

Program Code:

```
#include <stdio.h>

int main(){
    int i, j;
    int matrix[3][3];
    printf("Enter elements for a 3x3 matrix:\n");
    for (i = 0; i < 3; i++){
        for (j = 0; j < 3; j++){
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("\nTranspose:\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d ", matrix[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Enter elements for a 3x3 matrix:
1 2 3
4 5 6
7 8 9
Transpose:
1 4 7
2 5 8
3 6 9
```

Experiment 23

Aim: Write a program to find the factorial of a number by using the concept of recursion.

Program Code:

```
#include <stdio.h>

int factorial(int n);

int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Factorial of %d = %d\n", n, factorial(n));
    return 0;
}

int factorial(int n) {
    if (n >= 1) {
        return n * factorial(n - 1);
    } else {
        return 1;
    }
}
```

Output:

```
Enter a positive integer: 5
Factorial of 5 = 120
```


Experiment 24

Aim: Write a menu driven C program to show the use of in-built string functions like `strlen`, `strcat`, `strcpy`, `strcmp`, `strrev` etc.

Program Code:

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

void reverseString(char *str) {
    int length = strlen(str);
    for (int i = 0; i < length / 2; i++) {
        char temp = str[i];
        str[i] = str[length - i - 1];
        str[length - i - 1] = temp;
    }
}

int main() {
    char str1[100], str2[100];
    int choice;

    printf("Enter a string: ");
    fgets(str1, 100, stdin);
    printf("Menu:\n");
    printf("1. Find string length\n");
    printf("2. Concatenate two strings\n");
    printf("3. Copy a string\n");
    printf("4. Compare two strings\n");
    printf("5. Reverse a string\n");
    printf("Enter your choice: ");
```

```
scanf("%d", &choice);
switch (choice) {
    case 1:
        printf("Length of string: %d\n", strlen(str1));
        break;
    case 2:
        printf("Enter another string: ");
        getchar();
        fgets(str2, 100, stdin);
        strcat(str1, str2);
        printf("Concatenated string: %s\n", str1);
        break;
    case 3:
        strcpy(str2, str1);
        printf("Copied string: %s\n", str2);
        break;
    case 4:
        printf("Enter another string: ");
        getchar();
        fgets(str2, 100, stdin);
        if (strcmp(str1, str2) == 0) {
            printf("Strings are equal.\n");
        } else {
            printf("Strings are not equal.\n");
        }
        break;
    case 5:
```

```
reverseString(str1);  
printf("Reversed string: %s\n", str1);  
break;  
default:  
    printf("Invalid choice.\n");  
}  
return 0;  
}
```

Output:

```
Enter a string: My name is Kanav  
Menu:  
1. Find string length  
2. Concatenate two strings  
3. Copy a string  
4. Compare two strings  
5. Reverse a string  
Enter your choice: 5  
Reversed string:  
vanaK si eman yM
```

Experiment 25

Aim: Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string.

Program Code:

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

int main() {
    char str[100], substr[100];
    printf("Enter a string: ");
    fgets(str, 100, stdin);
    str[strcspn(str, "\n")] = '\0';
    printf("Enter a substring: ");
    fgets(substr, 100, stdin);
    substr[strcspn(substr, "\n")] = '\0';
    int count = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (strstr(&str[i], substr) == &str[i]) {
            count++;
        }
    }
    printf("Total occurrences of '%s' in '%s': %d\n", substr, str, count);
    return 0;
}
```

Output:

```
Enter a string: KKAANNAAVV
Enter a substring: AA
Total occurrences of 'AA' in 'KKAANNAAVV': 2
```

Experiment 26

Aim: Write a program to display the sum of the digits of a number by using the concept of recursion.

Program Code:

```
#include <stdio.h>

int sumOfDigits(int n);

int main() {
    int num;

    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Sum of digits of %d = %d\n", num, sumOfDigits(num));
    return 0;
}

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

Output:

```
Enter a positive integer: 5753
Sum of digits of 5753 = 20
```

Experiment 27

Aim: Write a C program to add two distances in inch & feet using the concept of structures.

Program Code:

```
#include <stdio.h>

struct Distance {
    int feet;
    float inch;
};

int main() {
    struct Distance d1, d2, sum;
    printf("Enter information for 1st distance:\n");
    printf("Enter feet: ");
    scanf("%d", &d1.feet);
    printf("Enter inch: ");
    scanf("%f", &d1.inch);
    printf("\nEnter information for 2nd distance:\n");
    printf("Enter feet: ");
    scanf("%d", &d2.feet);
    printf("Enter inch: ");
    scanf("%f", &d2.inch);
    sum.feet = d1.feet + d2.feet;
    sum.inch = d1.inch + d2.inch;
    if (sum.inch >= 12.0) {
        sum.inch = sum.inch - 12.0;
        ++sum.feet;
    }
}
```

```
printf("\nSum of distances = %d\'-%.1f\n", sum.feet, sum.inch);  
return 0;  
}
```

Output:

```
Enter information for 1st distance:  
Enter feet: 6  
Enter inch: 2  
  
Enter information for 2nd distance:  
Enter feet: 5  
Enter inch: 9  
  
Sum of distances = 11'-11.0
```

Experiment 28

Aim: Write a C program to add two complex numbers using the concept of structures in C.

Program Code:

```
#include <stdio.h>

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

complex add(complex n1, complex n2);

int main() {
    complex n1, n2, result;

    printf("For 1st complex number \n");
    printf("Enter the real and imaginary parts: ");
    scanf("%f %f", &n1.real, &n1.imag);
    printf("\nFor 2nd complex number \n");
    printf("Enter the real and imaginary parts: ");
    scanf("%f %f", &n2.real, &n2.imag);

    result = add(n1, n2);

    printf("Sum = %.1f + %.1fi\n", result.real, result.imag);
    return 0;
}
```



```
}  
  
complex add(complex n1, complex n2) {  
    complex temp;  
    temp.real = n1.real + n2.real;  
    temp.imag = n1.imag + n2.imag;  
    return (temp);  
}
```

Output:

```
For 1st complex number  
Enter the real and imaginary parts: 1.1  
-2.2  
  
For 2nd complex number  
Enter the real and imaginary parts: 3.3  
-4.4  
Sum = 4.4 + -6.6i
```

Experiment 29

Aim : Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure.

Program Code:

```
#include <stdio.h>

struct employee {
    char name[50];
    int empId;
    int age;
    float salary;
};

int main() {
    struct employee employees[5];
    for (int i = 0; i < 5; i++) {
        printf("Enter employee %d details:\n", i + 1);
        printf("Name: ");
        scanf("%s", employees[i].name);
        printf("Employee ID: ");
        scanf("%d", &employees[i].empId);
        printf("Age: ");
        scanf("%d", &employees[i].age);
        printf("Salary: ");
        scanf("%f", &employees[i].salary);
    }
    printf("\n");
    for (int i = 0; i < 5; i++) {
        printf("Employee %d details:\n", i + 1);
```

```
printf("Name: %s\n", employees[i].name);  
printf("Employee ID: %d\n", employees[i].empId);  
printf("Age: %d\n", employees[i].age);  
printf("Salary: %.2f\n", employees[i].salary);  
}  
return 0;  
}
```

Output:

```
Enter employee 1 details:  
Name: Kanav  
Employee ID: 2111  
Age: 19  
Salary: 50000  
Enter employee 2 details:  
Name: Mridul  
Employee ID: 2126  
Age: 18  
Salary: 40000  
Enter employee 3 details:  
Name: Kannan  
Employee ID: 2112  
Age: 18  
Salary: 45000  
Enter employee 4 details:  
Name: Kunal  
Employee ID: 2130  
Age: 18  
Salary: 42000  
Enter employee 5 details:  
Name: Adish  
Employee ID: 2090  
Age: 18  
Salary: 47000
```

```
Employee 1 details:  
Name: Kanav  
Employee ID: 2111  
Age: 19  
Salary: 50000.00  
Employee 2 details:  
Name: Mridul  
Employee ID: 2126  
Age: 18  
Salary: 40000.00  
Employee 3 details:  
Name: Kannan  
Employee ID: 2112  
Age: 18  
Salary: 45000.00  
Employee 4 details:  
Name: Kunal
```

```
Employee ID: 2130  
Age: 18  
Salary: 42000.00  
Employee 5 details:  
Name: Adish  
Employee ID: 2090  
Age: 18  
Salary: 47000.00
```

Experiment 30

Aim: Write a Program in C to store and retrieve the information about students of a university by using the concept of file handling.

Program Code:

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

struct student {
    char name[50];
    int roll;
    float marks;
};

int main() {
    struct student s;
    FILE *fptr;

    fptr = fopen("student.txt", "w");
    if (fptr == NULL) {
        printf("Error!");
        exit(1);
    }

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

```
fwrite(&s, sizeof(struct student), 1, fptr);  
fclose(fptr);  
fptr = fopen("student.txt", "r");  
if (fptr == NULL) {  
    printf("Error!");  
    exit(1);  
}  
while (fread(&s, sizeof(struct student), 1, fptr) == 1) {  
    printf("Name: %s\nRoll number: %d\nMarks: %.2f\n", s.name, s.roll, s.marks);  
}  
fclose(fptr);  
return 0;  
}
```

Output:

```
Enter name: Kanav  
Enter roll number: 2111  
Enter marks: 90  
Name: Kanav  
Roll number: 2111  
Marks: 90.00
```

Experiment 31

Aim: Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string.

Program Code:

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

int main() {
    char filename[100], search[100], replace[100], tempFilename[] = "temp.txt";
    printf("Enter the filename: ");
    scanf("%99s", filename);
    printf("Enter the search string: ");
    scanf("%99s", search);
    printf("Enter the replace string: ");
    scanf("%99s", replace);
    FILE *fptr = fopen(filename, "r");
    if (fptr == NULL) {
        printf("File not found!");
        exit(1);
    }
    FILE *temp = fopen(tempFilename, "w");
    if (temp == NULL) {
        printf("Error creating temporary file!");
        exit(1);
    }
    int count = 0;
```

```
char word[100];
while (fscanf(fptr, "%99s", word) == 1) {
    if (strcmp(word, search) == 0) {
        fputs(replace, temp);
        fputs(" ", temp);
        count++;
    } else {
        fputs(word, temp);
        fputs(" ", temp);
    }
}
fclose(fptr);
fclose(temp);
remove(filename);
rename(tempFilename, filename);
printf("Total occurrences of '%s' replaced with '%s': %d\n", search, replace, count);
return 0;
}
```

Output:

```
Enter the filename: text.txt
Enter the search string: Kanav
Enter the replace string: KanavKumar
Total occurrences of 'Kanav' replaced with 'KanavKumar': 1
```

 text.txt
1 KanavKumar