

**FAISAL FIRDOUS**

(Assistant Professor, JUIT Solan)

## C Programming

### Contents

1. Basics of C Programming [18 Programs]
2. Operators and Expressions [18]
3. Control Statements [27]
4. Patterns[15]
5. Functions[17]
6. Recursion[23]
7. Arrays [40]
8. Pointers [14]
9. Strings [20]
10. Structure and Union [17]
11. Files [7]
12. Sorting [7]

LinkedIn: <https://www.linkedin.com/in/faisal-firdous-1187301b4/>

### Free Resources to learn C Programming

1. Introduction to C Programming ([Introduction to C and C++ | Electrical Engineering and Computer Science | MIT OpenCourseWare](#))
2. (BEST) CS50 Course by Harvard ([Week 0 - CS50](#))
3. Free Interactive C Tutorial (<https://www.learn-c.org/>)

4. Let Us C Books ( <https://shorturl.at/nwxDT>)
5. C Programming HandWritten Notes Link: [C\\_Prog\\_Handwritten\\_Notes.pdf](#)
6. C Programming Projects: [C Programming Projects](#)
7. 100 C Programs For Placements: [100 C Programs for Placements](#)
8. 1000 C Programming Questions: [scribd.vpdfs.com\\_1000-c-programming-questions.pdf](#)

## Master C Programming in 30 days with free resources

### Week 1: Basics

1. Days 1-3: Learn the basics of C syntax, data types, and variables.
2. Days 4-7: Study control structures like loops (for, while) and conditional statements (if, switch).

### Week 2: Functions and Arrays

3. Days 8-10: Understand functions, how to create them, and pass parameters.
4. Days 11-14: Dive into arrays and how to manipulate them.

### Week 3: Pointers and Memory Management

5. Days 15-17: Learn about pointers and their role in C programming.
6. Days 18-21: Study memory management, dynamic memory allocation, and deallocation (malloc, free).

### Week 4: File Handling and Advanced Topics

7. Days 22-24: Explore file handling and I/O operations in C.
8. Days 25-28: Learn about more advanced topics like structures, unions, and advanced data structures.
9. Days 29-30: Practice and review what you've learned. Work on small projects to apply your knowledge.

Throughout the 30 days, make sure to:

- Code every day to reinforce your learning.
- Use online resources, tutorials, and textbooks.
- Join C programming communities and forums for help and discussions.

- Solve coding challenges and exercises to test your skills (e.g., HackerRank, LeetCode).
- Document your progress and make notes.

## Basics of C Programming

**//Program 1: WAP to print Hello World**

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

**//Program 2: WAP to print Name, rollno, Batch and Hometown using multiple printf functions**

```
#include<stdio.h>
int main()
{
    printf("My name is Mr. Faisal Firdous");
    printf("My rollno is 123");
    printf("My batch is 24A17");
    printf("I am from Jammu & Kashmir");
    return 0;
}
```

**//Program 3: WAP to print Name, rollno, Batch and Hometown using multiple printf functions with newline**

```
#include<stdio.h>
int main()
{
    printf("My name is Mr. Faisal Firdous\n");
    printf("My rollno is 123\n");
    printf("My batch is 24A17\n");
    printf("I am from Jammu & Kashmir");
    return 0;
}
```

**//Program 4: WAP to print Name, rollno, Batch and Hometown using single printf functions with newline**

```
#include<stdio.h>
int main()
{
    printf("My name is Mr. Faisal Firdous\nMy rollno is 123\nMy batch is 24A17\nI am from Jammu & Kashmir");
}
```

```

    return 0;
}
//Program 5: WAP to print the value of a variable
#include<stdio.h>
int main()
{
    int a=10;
    printf("The value of a =%d",a);
    return 0;
}
//Program 6: WAP to ask the user to enter the integer value of a variable and then print it
#include<stdio.h>
int main()
{
    int a;
    printf("Enter the value of a : ");
    scanf("%d",&a);
    printf("The value of a =%d",a);
    return 0;
}
//Program 7: WAP to ask the user to enter the character value of a variable and then print it
#include<stdio.h>
int main()
{
    char a;
    printf("Enter the value of a : ");
    scanf("%c",&a);
    printf("The value of a =%c",a);
    return 0;
}
//Program 8: WAP to ask the user to enter the float value and then print it
#include<stdio.h>
int main()
{
    float a;
    printf("Enter the value of a : ");
    scanf("%f",&a);
    printf("The value of a =%f",a);
    return 0;
}
//Program 9: WAP to ask the user to enter the float value and then print upto 2 decimal
#include<stdio.h>
int main()
{

```

```

float a;
printf("Enter the value of a : ");
scanf("%f",&a);
printf("The value of a =%.2f",a);
return 0;
}
//Program 10: WAP to ask the user to enter the string value and then display it.
#include<stdio.h>
int main()
{
    char str[10];
    printf("Enter the string: ");
    scanf("%s",str);
    printf("Entered string is %s",str);
    return 0;
}
//Program 11:WAP to ask user enter the age and print it
#include<stdio.h>
int main()
{
    int age;
    printf("Enter your age: ");
    scanf("%d",&age);
    printf("Your age is : %d",age);
    return 0;
}
//Program 12:Method 1 WAP to ask user enter the two values a and b then display them
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter the value of a: ");
    scanf("%d",&a);
    printf("Enter the value of b: ");
    scanf("%d",&b);
    printf("The value of a =%d and b =%d",a,b);
    return 0;
}
//Program 13:Method 2 WAP to ask user enter the two values a and b then display them
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter the value of a and b: ");

```

```

scanf("%d%d",&a,&b);
printf("The value of a =%d and b =%d",a,b);
return 0;
}
//Program 14:WAP to ask user enter the DOB in DD-MM-YYYY format and display in the same format
#include<stdio.h>
int main()
{
    int dd,mm,yyyy;
    printf("Enter the DOB in DD-MM-YYYY format: ");
    scanf("%d-%d-%d",&dd,&mm,&yyyy);
    printf("Your DOB is %d-%d-%d",dd,mm,yyyy);
    return 0;
}
//Program 15:WAP to check Hexadecimal equivalent to decimal
#include<stdio.h>
int main()
{
    int num;
    printf("Enter the decimal value: ");
    scanf("%d",&num);
    printf("Decimal = %d and Hexadecimal = %x",num,num);
    return 0;
}
//Program 16:WAP to check Octal equivalent to decimal
#include<stdio.h>
int main()
{
    int num;
    printf("Enter the decimal value: ");
    scanf("%d",&num);
    printf("Decimal = %d and Octal = %o",num,num);
    return 0;
}
//Program 17:WAP to show the use of getchar() and putchar()
#include<stdio.h>
int main()
{
    char ch;
    printf("Enter the character: ");
    ch=getchar();
    printf("The entered character is : ");
    putchar(ch);

```

```

        return 0;
    }
//Program 18:WAP to show the concept of Symbolic Constant
#include<stdio.h>
#define MAX 100
#define PI 3.14159625
#define CH 'Y'
#define NAME "Faisal"

int main()
{
    printf("Max value is : %d\n", MAX);
    printf("PI value is : %lf\n", PI);
    printf("Character value is : %c\n", CH);
    printf("My NAME is : %s", NAME);

    return 0;
}

```

## Operators and Expressions

**//Program 1: WAP to understand the integer Arithmetic Operation**

```

#include<stdio.h>
int main()
{
    int a=19,b=5;
    printf("Sum =%d\n",a+b);
    printf("Difference =%d\n",a-b);
    printf("Multiplication =%d\n",a*b);
    printf("Division(quotient) =%d\n",a/b);
    printf("Modulus(Remainder) =%d\n",a%b);
    return 0;
}

```

**//Program 2: WAP to understand the floating point Arithmetic Operation**

```

#include<stdio.h>
int main()
{
    float a=13.5,b=5.8;
    printf("Sum =%.2f\n",a+b);
    printf("Difference =%.2f\n",a-b);
    printf("Multiplication =%.2f\n",a*b);
    printf("Division(quotient) =%.2f\n",a/b);
    return 0;
}

```

**//Program 3: WAP to understand the Assignment operator**

```

#include<stdio.h>

```

```

int main()
{
    int x=8,y=5,s=x+y+5;
    int a,b,c;
    a=b=c=30;
    printf("Value of x= %d\n",x);
    printf("Value of y= %d\n",y);
    printf("Value of s= %d\n",s);
    printf("Value of a= %d, b= %d and c = %d",a,b,c);
    return 0;
}

```

**//Program 4: WAP to understand the Prefix increment/decrement**

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

```

int main()
{
    int x=12;
    printf("x = %d\n",x);
    printf("++x = %d\n",++x);
    printf("x = %d\n",x);
    printf("--x = %d\n",--x);
    printf("x = %d\n",x);
    return 0;
}

```

**//Program 5: WAP to understand the Postfix increment/decrement**

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

```

int main()
{
    int x=15;
    printf("x = %d\n",x);
    printf("x++ = %d\n",x++);
    printf("x = %d\n",x);
    printf("x-- = %d\n",x--);
    printf("x = %d\n",x);
    return 0;
}

```

**//Program 6: WAP to understand the Relational Operators**

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

```

int main()
{
    int a,b;
    printf("Enter the value of a and b: ");
    scanf("%d%d",&a,&b);
    if(a<b)
        printf("%d is less than %d\n",a,b);
}

```



```

if(a<=b)
    printf("%d is less than and equal to %d\n",a,b);
if(a>b)
    printf("%d is greater than %d\n",a,b);
if(a>=b)
    printf("%d is greater than and equal to %d\n",a,b);
if(a==b)
    printf("%d is equal to %d\n",a,b);
if(a!=b)
    printf("%d is not equal to %d",a,b);

return 0;
}

```

**//Program 7:WAP to understand the sizeof operator**

```

#include<stdio.h>
int main( )
{
    int var=45;
    printf("Size of int =%d\n",sizeof(int));
    printf("Size of short =%d\n",sizeof(short));
    printf("Size of float =%d\n",sizeof(float));
    printf("Size of char =%d\n",sizeof(char));
    printf("Size of var = %d\n",sizeof(var));
    printf("Size of double = %d\n",sizeof(double));
    printf("Size of an integer constant = %d",sizeof(45));
    return 0;
}

```

**//Program 8: WAP to understand the use of comma operator**

```

#include<stdio.h>
int main( )
{
    int a,b,c,sum;
    sum =(a=8,b=7,c=9,a+b+c);
    printf("Sum =%d",sum);

    return 0;
}

```

**//Program 9: WAP to print the larger of two numbers using conditional operator**

```

#include<stdio.h>
int main( )
{
    int a,b,max;
    printf ("Enter the values a and b :");
    scanf("%d%d",&a,&b);

```

```

    max = a>b ? a :b;
    printf("Larger of %d and %d is %d",a,b,max);
    return 0;
}

```

**//Program 10. WAP to enter the temperature in Celsius and convert that into Fahrenheit.**

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

```

int main() {
    float celsius, fahrenheit;
    printf("Enter the temperature in Celsius: ");
    scanf("%f",&celsius);

    fahrenheit = (celsius * 9/5) + 32;
    printf("Temperature in Fahrenheit: %.2f\n", fahrenheit);

    return 0;
}

```

**//Program 11. WAP to accept the radius of the circle and calculate the area and perimeter of the circle.**

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

```
#define PI 3.14159
```

```

int main() {
    float radius, area, perimeter;
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    area = PI * radius * radius;
    perimeter = 2 * PI * radius;
    printf("Area of the circle: %.2f\n", area);
    printf("Perimeter (Circumference) of the circle: %.2f\n", perimeter);

    return 0;
}

```

**//Program 12. WAP to accept the number in decimal and print the number in octal and hexadecimal.**

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

```

int main() {
    int decimal;
    printf("Enter a decimal number: ");
    scanf("%d", &decimal);
    printf("Octal: %o\n", decimal);
    printf("Hexadecimal: %X\n", decimal);
}

```

```

    return 0;
}

```

**//Program 13. WAP to accept any five digit number and print the value of remainder after dividing by 3**

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

```

int main() {
    int number, remainder;

    printf("Enter a five-digit number: ");
    scanf("%d",&number);
    remainder = number%3;
    printf("The remainder when %d is divided by 3 is: %d\n", number, remainder);

    return 0;
}

```

**//Program 14. WAP to accept any two numbers, if the first number is greater than second then print the sum of these two numbers, otherwise print their difference using ternary operator.**

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

```

int main() {
    int num1, num2, result;
    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);

    result = (num1 > num2) ? (num1 + num2) : (num1 - num2);
    printf("The result is: %d\n", result);
    return 0;
}

```

**//Program 15. WAP to accept the principal, rate, and number of years and find out the simple interest.**

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

```

int main() {
    float principal, rate, years, simple_interest;
    printf("Enter the principal amount: ");
    scanf("%f", &principal);
    printf("Enter the rate of interest: ");
    scanf("%f", &rate);
    printf("Enter the number of years: ");
    scanf("%f", &years);
}

```

```

    simple_interest = (principal * rate * years) / 100;
    printf("The simple interest is: %.2f\n", simple_interest);

    return 0;
}
//Program 16: WAP that accepts marks in five subject and calculate the total percentage marks
#include <stdio.h>
int main() {
    float subject1, subject2, subject3, subject4, subject5, total, percentage;
    printf("Enter the marks for Subject 1: ");
    scanf("%f", &subject1);
    printf("Enter the marks for Subject 2: ");
    scanf("%f", &subject2);
    printf("Enter the marks for Subject 3: ");
    scanf("%f", &subject3);
    printf("Enter the marks for Subject 4: ");
    scanf("%f", &subject4);
    printf("Enter the marks for Subject 5: ");
    scanf("%f", &subject5);

    total = subject1 + subject2 + subject3 + subject4 + subject5;
    percentage = (total / 500) * 100;
    printf("Total Marks: %.2f\n", total);
    printf("Percentage: %.2f%%\n", percentage);
    return 0;
}
//Program 17: WAP to illustrate the use of Cast Operator
#include <stdio.h>
int main() {
    int x=5,y=2;
    float p,q;
    p=x/y;
    printf("p = %f\n",p);
    q=(float)x/y;
    printf("q = %f\n",q);
    return 0;
}
//Program 18: WAP to evaluate some expressions
#include<stdio.h>
int main ()
{
    int a,b,c,d,e,f,g,h,k;
    a=8,b=4,c=2,d=1,e=5,f=20;

```

```

printf("%d\t",a+b-(c+d)*3%e+f/9);
a=17,b=5,c=6,d=3,e=5;
printf("%d\t",a%6-b/2+(c*d-5)/e);
a=4,b=5,c=6,d=3,e=5,f=10;
printf("%d\t",a*b-c/d<e+f);
a=8,b=5,c=8,d=3,e=6,f=10,g=2,h=5,k=2;
printf("%d\t",a-b+c/d==e/f-g+h%k);
a=8,b=3,c=2,d=3,e=2,f=11;
printf("%d\n",a-b||(a-b*c)+d&&e-f%3);
return 0;
}

```

## Control Statements

**//Program 1:WAP to print a message if negative number is entered**

```

#include<stdio.h>
int main()
{
    int num;
    printf("Enter a num: ");
    scanf("%d",&num);
    if(num<0)
        printf("Number entered is negative.\n");
    printf("The number is %d ",num);
    return 0;
}

```

**//Program 2:WAP to print the larger and smaller of two number**

```

#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter value of a and b: ");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("%d is larger than %d",a,b);
    else
        printf("%d is larger than %d",b,a);
    return 0;
}

```

**//Program 3:WAP to find even and odd from entered number**

```

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

```

```

    if(num%2==0)
        printf("%d is Even.",num);
    else
        printf("%d is Odd.",num);

    return 0;
}
//Program 4: WAP to find the larger among 3 numbers
#include<stdio.h>
int main()
{
    int a,b,c,large;
    printf("Enter three number: ");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b) //1 2 3
    {
        if(a>c)
            large=a;
        else
            large=c;
    }
    else
    {
        if(b>c)
            large=b;
        else
            large=c;
    }
    printf("Larger among %d, %d ,%d is %d",a,b,c,large);
    return 0;
}
//Program 5: WAP to print values from 1 to 10 using for loop
#include<stdio.h>
int main()
{
    int i=1;
    while(i<=10)
    {
        printf("%d\t",i);
        i++;
    }
    return 0;
}
//Program 6:WAP to print the sum of digits of a number.

```

```
#include<stdio.h>
int main()
{
    int n,sum=0,rem;
    printf("Enter a number: ");
    scanf("%d",&n);
    int num=n;
    while(n>0){
        rem=n%10;
        sum=sum+rem;
        n=n/10;
    }
    printf("Sum of digits of a %d is %d",num,sum);
    return 0;
}
```

**//Program 7:WAP to print the product of digits of a number.**

```
#include<stdio.h>
int main()
{
    int n,prod=1,rem;
    printf("Enter a number: ");
    scanf("%d",&n);
    int num=n;
    while(n>0){
        rem=n%10;
        prod=prod*rem;
        n=n/10;
    }
    printf("Sum of digits of a %d is %d",num,prod);
    return 0;
}
```

**//Program 8:WAP to find the factorial of a number**

```
#include<stdio.h>
int main()
{
    int n,fact=1;
    printf("Enter a number: ");
    scanf("%d",&n);
    int num=n;
    if(n<0)
        printf("No Factorial of -ve number.");
    else{
        while(n>1)
        {
```

```

        fact = fact * n;
        n--;
    }
    printf("Factorial of %d is %d", num, fact);
}
return 0;
}

//Program 9: WAP to print the numbers from 10 to 20 using do while
#include <stdio.h>
int main()
{
    int i = 10;
    do
    {
        printf("%d\t", i);
        i++;
    } while(i <= 20);
    return 0;
}

//Program 10: WAP to count the digits of a number
#include <stdio.h>
int main()
{
    int count = 0, n;
    printf("Enter a number: ");
    scanf("%d", &n);
    do
    {
        n = n / 10;
        count++;
    } while(n > 0);
    printf("Number of digits = %d", count);
    return 0;
}

//Program 11: WAP to print numbers from 1 to 10
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i <= 10; i++)
    {
        printf("%d\t", i);
    }
}

```



```

    return 0;
}

```

**//Program 12:WAP to print even numbers from 2 to 20**

```

#include<stdio.h>
int main()
{
    int i;
    for(i=2;i<=20;i+=2)
    {
        printf("%d\t",i);

    }
    return 0;
}

```

**//Program 13:WAP to find the sum of this series 1+2+4+7+11+16+21+.....**

```

#include<stdio.h>
int main()
{
    int i,n,term=1,sum=0;
    printf("Enter number of terms: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        sum=sum+term;
        term=term+i;
    }
    printf("Sum = %d",sum);
    return 0;
}

```

**//Program 14:WAP to print the fibonacci series upto n terms 0 1 1 2 3 5 8 13.....**

```

#include<stdio.h>
int main()
{
    int x,y,z,n,i;
    x=0;
    y=1;
    printf("Enter number of terms: ");
    scanf("%d",&n);
    printf("%d\t%d\t",x,y);
    for(i=1;i<n;i++)
    {
        z=x+y;
        printf("%d\t",z);
    }
}

```

```

        x=y;
        y=z;
    }
    return 0;
}

```

**//Program 15: WAP to swap two numbers using temporary variable**

```

#include<stdio.h>
int main()
{
    int a,b,temp;
    printf("Enter value of a and b: ");
    scanf("%d%d",&a,&b);
    printf("Before swapping value of a =%d and b =%d\n",a,b);
    temp=a;
    a=b;
    b=temp;
    printf("After swapping value of a =%d and b =%d",a,b);

    return 0;
}

```

**//Program 16: WAP to swap two numbers without using temporary variable**

```

#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter value of a and b: ");
    scanf("%d%d",&a,&b);
    printf("Before swapping value of a =%d and b =%d\n",a,b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("After swapping value of a =%d and b =%d",a,b);

    return 0;
}

```

**//Program 17: WAP to check whether year is leap or not**

```

#include<stdio.h>
int main()
{
    int year;
    printf("Enter the year: ");
    scanf("%d",&year);
    if(year%100==0)
    {

```

```

        if(year%400==0)
            printf("Leap Year");
        else
            printf("Not Leap Year");
    }
else
{
    if(year%4==0)
        printf("Leap Year");
    else
        printf("Not Leap Year");
}
return 0;
}

//Program 18: WAP to find the factorial of a number
#include<stdio.h>
int main()
{

    int n,fact=1,num;
    printf("Enter a number: ");
    scanf("%d",&n);
    num=n;
    if(n<0)
        printf("No factorial of a negative number.");
    else
    {
        while(n>1){
            fact=fact*n;
            n--;
        }
        printf("Factorial of a %d = %d",num,fact);
    }
    return 0;
}

//Program19: WAP to print Armstrong numbers from 100 to 999
#include<stdio.h>
int main()
{
    int i,num,n,cube,sum,d;
    printf("Armstrongs numbers are: ");
    for(i=100;i<=999;i++)
    {
        n=i;

```

```

    sum=0;
    while(n>0){
        d=n%10;
        n=n/10;
        cube=d*d*d;
        sum=sum+cube;
    }
    if(i==sum)
        printf("%d\n",i);
}
return 0;
}
//Program 20: WAP to check the number is perfect or not
#include <stdio.h>
int main ()
{
    int num = 28, sum = 0;
    // iteratively check for all numbers in range [1, 27]
    for(int i = 1; i < num; i++){
        // check if i is a divisor, if yes then add to sum
        if(num % i == 0)
            sum = sum + i;
    }
    if(sum == num)
        printf("%d is a perfect number",num);
    else
        printf("%d is not a perfect number",num);
}

```

## Pattern Programs

```

//Program :
// P1 *****
#include<stdio.h>
int main()
{
    int i,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
        printf("* ");
    return 0;
}
//Program :

```

```

/* p2
*****
*****
*****
*****
*/
#include<stdio.h>
int main()
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        for(j=0;j<5;j++)
            printf("* ");
        printf("\n");
    }
    return 0;
}

```

**//Program :**

```

/* p3
*
**
***
****
*****
*/
#include<stdio.h>
int main()
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("* ");
        printf("\n");
    }

    return 0;
}

```

**//Program :**

/\* P4

1  
22  
333  
4444  
55555  
\*/

#include<stdio.h>

int main()

```
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("%d ",i);
        printf("\n");
    }

    return 0;
}
```

**//Program :**

/\* P5

1  
12  
123  
1234  
12345  
\*/

#include<stdio.h>

int main()

```
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("%d ",j);
        printf("\n");
    }

    return 0;
}
```

**//Program :**

```
/* P6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
*/
#include<stdio.h>
int main()
{
    int i,j,n,p=0;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("%d ",++p);
        printf("\n");
    }
    return 0;
}
```

**//Program :**

```
/* P7
2
3 4
4 5 6
5 6 7 8
6 7 8 9 10
*/
#include<stdio.h>
int main()
{
    int i,j,n;
    printf("Enter th value of n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        {
            for(j=1;j<=i;j++)
                printf("%d ",i+j);
            printf("\n");
        }
    }
}
```

```
    return 0;
}
```

**//Program :**

/\* P8

1

0 1

1 0 1

0 1 0 1

1 0 1 0 1

\*/

#include<stdio.h>

int main()

{

int i,j,n;

printf("Enter the value of n: ");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

for(j=1;j<=i;j++)

{

if((i+j)%2==0)

printf("1 ");

else

printf("0 ");

}

printf("\n");

}

return 0;

}

**//Program :**

/\* P9

5 4 3 2 1

5 4 3 2

5 4 3

5 4

5

\*/

#include<stdio.h>



```

int main()
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n); //n=5

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

    return 0;
}

```

**//Program :**

/\* P10

5

4 4

3 3 3

2 2 2 2

1 1 1 1 1

\*/

#include<stdio.h>

int main()

```

{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n); //n=5

```

```

    return 0;
}

```

**//Program :**

/\* P11

\* \* \* \* \*

\* \* \* \*

```
* * *
```

```
* *
```

```
*
```

```
*/
```

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

```
int main()
```

```
{
```

```
    int i,j,n;
```

```
    printf("Enter the value of n: ");
```

```
    scanf("%d",&n); //n=5
```

```
    for(i=n;i>=1;i--)
```

```
    {
```

```
        for(j=1;j<=i;j++)
```

```
            printf("* ");
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

**//Program :**

```
/* P12
```

```
5 5 5 5 5
```

```
4 4 4 4
```

```
3 3 3
```

```
2 2
```

```
1
```

```
*/
```

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

```
int main()
```

```
{
```

```
    int i,j,n;
```

```
    printf("Enter the value of n: ");
```

```
    scanf("%d",&n); //n=5
```

```
    for(i=n;i>=1;i--)
```

```
    {
```

```
        for(j=1;j<=i;j++)
```

```
            printf("%d ",i);
```

```
        printf("\n");
```

```
}
```

```

    return 0;
}

```

**//Program :**

/\* P13

```

1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

```

```

*/
#include<stdio.h>
int main()
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n); //n=5
    for(i=n;i>=1;i--)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
}

```

```

    return 0;
}

```

**//Program :**

/\* P14

```

5 4 3 2 1
5 4 3 2
5 4 3
5 4
5

```

```

*/
#include<stdio.h>
int main()

```

```

{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n); //n=5

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

    return 0;
}

```

**//Program :**

/\* P15

```

1 1 1 1 1
2 2 2 2
3 3 3
4 4
5

```

```

*/
#include<stdio.h>
int main()
{
    int i,j,n;
    printf("Enter the value of n: ");
    scanf("%d",&n); //n=5
    for(i=n;i>=1;i--)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d ",n+1-i);
        }
        printf("\n");
    }

    return 0;
}

```

```

}
//Program 21: WAP to print whether the number is even or odd using goto
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the number :");
    scanf("%d",&n);
    if(n%2==0)
        goto even;
    else
        goto odd;
    even:
        printf("Number is even");
        goto end;
    odd:
        printf("Number is odd");
        goto end;
    end:

    return 0;
}

```

## Switch Control Statement

```

//Program 22:WAP to understand the switch control statement
#include<stdio.h>
int main()
{
    int num;
    printf("Enter the value of num: ");
    scanf("%d",&num);
    switch(num) //1
    {
        case 1:
            printf("First \n");
        case 2:
            printf("Second \n");
        case 3:
            printf("Third \n");
        default:
            printf("Wrong choice and enter number between 1-3:");
    }
}

```

```

    }

    return 0;
}

```

**//Program 23:WAP to understand the switch control statement using break**

```

#include<stdio.h>
int main()
{
    int num;
    printf("Enter the value of num: ");
    scanf("%d",&num);
    switch(num) //1
    {
        case 1:
            printf("First \n");
            break;
        case 2:
            printf("Second \n");
            break;
        case 3:
            printf("Third \n");
            break;
        case 4:
            printf("Fourth \n");
            break;
        default:
            printf("Wrong choice and enter number between 1-4:");
    }
    return 0;
}

```

**//Program 24:WAP to perform arithmetic calculations on integers**

```

#include<stdio.h>
int main()
{
    char op;
    int a,b;
    printf("Enter number operator and other number: ");
    scanf("%d%c%d",&a,&op,&b);
    switch(op)
    {
        case '+':
            printf("Result =%d\n",a+b);
            break;
        case '-':

```

```

        printf("Result =%d\n",a-b);
        break;
case '/':
    printf("Result =%d\n",a/b);
    break;
case '*':
case 'x':
case 'X':
    printf("Result =%d\n",a*b);
    break;
case '%':
    printf("Result =%d\n",a%b);
    break;
default:
    printf("Enter valid operator.");

}

return 0;
}

```

**//Program 25:WAP to check whether the alphabet is a vowel or consonant**

```

#include<stdio.h>
int main()
{
    char ch;
    printf("Enter an alphabet: ");
    scanf("%c",&ch);
    switch(ch)
    {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            printf("Alphabet is Vowel \n");
            break;
        default:
            printf("Alphabet is consonant");

    }
    return 0;
}

```

**//Program 26:WAP to create a menu driven program using infinite loop and switch**

**/\***

**1. Create database**

**2. Insert new records**

**3. Modify a record**

**4. Delete a record**

**5. Display a record**

**6. exit**

**\*/**

**#include<stdio.h>**

**int main()**

**{**

int choice;

while(1)

{

printf("1. Create database \n");

printf("2. Insert new records \n");

printf("3. Modify a record \n");

printf("4. Delete a record \n");

printf("5. Display a record \n");

printf("6. exit \n\n");

printf("Enter your choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("Database Created.....\n");

break;

case 2:

printf("Record Inserted.....\n");

break;

case 3:

printf("Record modified.....\n");

break;

case 4:

printf("Record deleted.....\n");

break;

case 5:

printf("Records displayed.....\n");

break;

case 6:

exit(1);

default:



```

        printf("Wrong choice...!");
    }
}

return 0;
}
//Program 27:WAP to print the days of a week
#include<stdio.h>
int main()
{
    int day;
    printf("Enter the value of day: ");
    scanf("%d",&day);
    switch(day)
    {
    case 1:
        printf("Monday");
        break;
    case 2:
        printf("Tuesday");
        break;
    case 3:
        printf("Wednesday");
        break;
    case 4:
        printf("Thrusday");
        break;
    case 5:
        printf("Friday");
        break;
    case 6:
        printf("Saturday");
        break;
    case 7:
        printf("Sunday");
        break;
    default:
        printf("Enter choice between 1-7: ");
    }

    return 0;
}

```

## Functions

**//Program 1: WAP to find the square root using Library Functions**

```
#include<stdio.h>
#include<math.h>
int main()
{
    double n,s;
    printf("Enter a number: ");
    scanf("%lf",&n);
    s=sqrt(n);
    printf("The square root of %.2lf is %.2lf",n,s);
    return 0;
}
```

**//Program 2: WAP to find the sum of two numbers**

```
#include<stdio.h>
int sum(int x,int y);
int main()
{
    int a,b,s;
    printf("Enter the values of a and b: ");
    scanf("%d %d",&a,&b);
    s=sum(a,b);
    printf("Sum of %d and %d is %d",a,b,s);

    return 0;
}
int sum(int x,int y)
{
    int s;
    s=x+y;
    return s;
}
```

**//Program 3: WAP to print message using user-defined functions**

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

    return 0;
}
```

```
void display()
{
    printf("Data Structure\n");
}
```

**//Program 4: WAP to check whether number is Even or Odd**

```
#include<stdio.h>
void find(int n);
int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d",&num);
    find(num);
    return 0;
}
void find(int n)
{

    if(n%2==0)
        printf("%d is even",n);
    else
        printf("%d is odd",n);
}
```

**//Program 5: WAP that finds larger among two numbers**

```
#include<stdio.h>
int max(int a,int b);
int main()
{
    int a,b;
    printf("Enter value of a and b: ");
    scanf("%d%d",&a,&b);
    printf("Larger among %d and %d is %d",a,b,max(a,b));
    return 0;
}
int max(int a,int b)
{
    if(a>b)
        return a;
    else
        return b;
}
```

**//Program 6: WAP to find the factorial of a number**

```
#include<stdio.h>
long int factorial(int n);
```

```

int main()
{
    int num;
    printf("Enter a number : ");
    scanf("%d",&num);
    if(num<0)
        printf("No factorial of negative number");
    else
        printf("Factorial of %d is %d",num,factorial(num));

    return 0;
}
long int factorial(int n)
{
    int i,fact=1;
    if(n==0)
        return 1;
    for(i=n;i>1;i--)
        fact=fact*i;
    return fact;
}

```

**//Program 7: WAP to find the sum of squares of all the odd numbers from 1 to 25**

```

#include<stdio.h>
int sumOfOdd(void);
int main()
{
    printf("Sum of all odd numbers from 1 - 25 is %d ",sumOfOdd());

    return 0;
}
int sumOfOdd()
{
    int i,sum=0;
    for(i=1;i<=25;i++)
        if(i%2!=0)
            sum+=i*i;
    return sum;
}

```

**//Program 8: WAP to find the sum of digits of a number**

```

#include<stdio.h>
int sumOfDigits(int n);
int main()
{
    int num;

```

```

    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Sum of digits of %d is %d ",num,sumOfDigits(num));

    return 0;
}
int sumOfDigits(int n)
{
    int sum=0,rem;
    while(n>0)
    {
        rem=n%10;
        sum=sum+rem;
        n=n/10;
    }
    return sum;
}
//Program 9: WAP to check the reverse and whether number is palindrome or not.
#include<stdio.h>
int reverse(int n);
int main()
{
    int num;
    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Reverse of %d is %d \n",num,reverse(num));
    if(num==reverse(num))
        printf("%d is palindrome",num);
    else
        printf("%d is not palindrome",num);

    return 0;
}
int reverse(int n)
{
    int rev=0,rem;
    while(n>0)
    {
        rem=n%10;
        rev=rev*10+rem;
        n=n/10;
    }
    return rev;
}

```

**//Program 10: WAP to check whether number is prime or not**

```

#include<stdio.h>
#include<math.h>
int prime(int n);
int main()
{
    int num;
    printf("Enter the number: ");
    scanf("%d",&num);
    if(prime(num))
        printf("%d is prime",num);
    else
        printf("%d is not prime",num);

    return 0;
}
int prime(int n)
{
    int flag=1,i;
    for(i=2;i<sqrt(n);i++)
    {
        if(n%i==0)
        {
            flag=0;
            break;
        }
    }

    return flag;
}

```

**//Program 11: WAP to find prime numbers upto 500**

```

#include<stdio.h>
#include<math.h>
int isPrime(int n);
int main()
{
    int i;
    for(i=0;i<500;i++)
        if(isPrime(i))
            printf("%d \t",i);

    return 0;
}
int isPrime(int n)

```

```

{
    int flag=1,i;
    for(i=2;i<sqrt(n);i++)
    {
        if(n%i==0)
        {
            flag=0;
            break;
        }
    }

    return flag;
}

```

**//Program 12: WAP to convert decimal into binary number**

```

#include<stdio.h>
int binary(int n);
int main()
{
    int num;
    printf("Enter a decimal number: ");
    scanf("%d",&num);
    printf("Decimal =%d and Binary =%d",num,binary(num));

    return 0;
}
int binary(int n)
{
    int a=1,rem,bin=0;
    while(n>0)
    {
        rem=n%2;
        bin=bin+rem*a;
        n=n/2;
        a=a*10;
    }
    return bin;
}

```

**Program 13 WAP to convert a binary or octal number to a decimal number depending on user's choice**

**Program 14 WAP to implement these formulae of permutations and combinations.**

**Program 15 WAP to print Pascal's triangle.**

**Program 16 WAP to convert a decimal number to a roman number**

**Program 17** Write a function that accepts a character in lower case and returns its upper case equivalent.

## Arrays

**//Program 1: WAP to take elements of an array and display them.**

```
#include<stdio.h>
int main()
{   int i,arr[5];
    for(i=0;i<5;i++)
    {
        printf("Enter the element arr[%d]= ",i);
        scanf("%d",&arr[i]);
    }
    for(i=0;i<5;i++)
        printf("The element at arr[%d] = %d\n",i,arr[i]);

    return 0;
}
```

**//Program 2: WAP to print 1D array**

```
#include <stdio.h>
int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int i;
    printf("Array Elements are: ");
    for(i = 0; i < 10; i++)
        printf("%d ", array[i]);

    return 0;
}
```

**//Program 3: WAP to Declare, Access and process Array**

```
#include<stdio.h>
#define SIZE 10
int main()
{
    int arr[10];
```



```

//read
for(int i=0;i<10;i++)
    scanf("%d",&arr[i]);
//display
for(int i=0;i<10;i++)
    printf("%d",arr[i]);
return 0;
}

```

**//Program 4: WAP to print reverse array**

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

```

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int i;
    printf("Reverse Elements are: ");
    for(i = 9; i >= 0; i--)
        printf("%d ", array[i]);
    return 0;
}

```

**//Program 5: WAP to calculate average of array**

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

```

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int sum, i;
    float avg;
    sum = avg = 0;
    for(i = 0; i < 10; i++) {
        sum = sum + array[i];
    }

    avg = (float)sum / i;
    printf("Average of array values is %.2f", avg);

    return 0;
}

```

**//Program 6: WAP to find largest array element**

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

```

int main() {
    int array[10] = {11, 24, 32, 42, 5, 6, 71, 8, 9, 0};
    int i, largest;
    largest = array[0];
    for(i = 1; i < 10; i++) {
        if( largest < array[i] )
            largest = array[i];
    }
}

```

```

    }
    printf("Largest element of array is %d", largest);
    return 0;
}

//Program 7: WAP to find the Second largest array element
#include <stdio.h>

int main() {
    int array[10] = {101, 11, 3, 4, 50, 69, 7, 8, 9, 0};
    int i, largest, second;

    if(array[0] > array[1]) {
        largest = array[0];
        second = array[1];
    } else {
        largest = array[1];
        second = array[0];
    }

    for(i = 2; i < 10; i++) {
        if( largest < array[i] ) {
            second = largest;
            largest = array[i];
        } else if( second < array[i] ) {
            second = array[i];
        }
    }

    printf("Largest - %d \n Second - %d \n", largest, second);

    return 0;
}

```

**//Program 8: WAP to find smallest array element**  
#include <stdio.h>

```

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int i, smallest;

    smallest = array[0];

    for(i = 1; i < 10; i++) {
        if( smallest > array[i] )
            smallest = array[i];
    }

    printf("Smallest element of array is %d", smallest);
}

```

```

    return 0;
}
//Program 9: WAP to copy array elements
#include <stdio.h>

int main() {
    int original[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int copied[10];
    int i;

    for(i = 0; i < 10; i++) {
        copied[i] = original[i];
    }
    printf("original -> copied \n");

    for(i = 0; i < 10; i++) {
        printf("  %2d    %2d\n", original[i], copied[i]);
    }

    return 0;
}
//Program 10: WAP to reverse copy array elements
#include <stdio.h>

int main() {
    int original[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int copied[10];
    int i, count;
    count = 9;
    for(i = 0; i < 10; i++) {
        copied[count] = original[i];
        count--;
    }
    printf("original -> copied \n");
    for(i = 0; i < 10; i++) {
        printf("  %2d %2d\n", original[i], copied[i]);
    }

    return 0;
}

//Program 11: WAP to add the elements of an array
#include<stdio.h>
int main()

```

```

{
    int arr[10],i,sum=0;
    for(i=0;i<10;i++)
    {
        printf("Enter the value of arr[%d] :",i);
        scanf("%d",&arr[i]);
        sum=sum+arr[i];
    }
    printf("Sum =%d",sum);
    return 0;
}

```

**//Program 12: WAP to count the even and odd numbers in a array**

```

#include<stdio.h>
#define SIZE 10
int main()
{
    int arr[SIZE],i,even=0,odd=0;
    for(i=0;i<SIZE;i++)
    {
        printf("Enter the value of arr[%d] :",i);
        scanf("%d",&arr[i]);
        if(arr[i]%2==0)
            even++;
        else
            odd++;
    }
    printf("Even numbers = %d and odd numbers %d",even,odd);

    return 0;
}

```

**//Program 13: WAP to find maximum and minimum element in an array**

```

#include<stdio.h>
int main()
{
    int i,arr[5]={7,2,11,4,5};
    int min,max;
    min=max=arr[0];
    for(i=1;i<5;i++)
    {
        if(arr[i]<min)
            min=arr[i];
        if(arr[i]>max)
            max=arr[i];
    }
}

```

```

    }
    printf("Min= %d and Max = %d",min,max);

```

```

    return 0;
}

```

**//Program 14: WAP to find Even and Odd numbers in an Array**

```

#include <stdio.h>

```

```

int main() {
    int array[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    int even[10], odd[10];
    int loop, e, d;

```

```

    e = d = 0;
    for(loop = 0; loop < 10; loop++) {
        if(array[loop]%2 == 0) {
            even[e] = array[loop];
            e++;
        } else {
            odd[d] = array[loop];
            d++;
        }
    }
}

```

```

printf(" original -> ");

```

```

for(loop = 0; loop < 10; loop++)
    printf(" %d", array[loop]);
printf("\n even -> ");
for(loop = 0; loop < e; loop++)
    printf(" %d", even[loop]);
printf("\n odd -> ");
for(loop = 0; loop < d; loop++)
    printf(" %d", odd[loop]);
return 0;
}

```

**//Program 15: WAP to concatenate two arrays**

```

#include <stdio.h>

```

```

int main() {
    int array[10];
    int even[5] = {0, 2, 4, 6, 8};
    int odd[5] = {1, 3, 5, 7, 9};
    int loop, index, e_len, o_len;
    e_len = o_len = 5;
    index = 0;

```

```

for(loop = 0; loop < e_len; loop++) {
    array[index] = even[loop];
    index++;
}
for(loop = 0; loop < o_len; loop++) {
    array[index] = odd[loop];
    index++;
}
printf("\nEven -> ");

for(loop = 0; loop < e_len; loop++)
    printf(" %d", even[loop]);
printf("\nOdd -> ");
for(loop = 0; loop < o_len; loop++)
    printf(" %d", odd[loop]);
printf("\nConcat -> ");
for(loop = 0; loop < 10; loop++)
    printf(" %d", array[loop]);

return 0;
}

```

**//Program 16: WAP to search the element in an array using Linear Search**

```

#include<stdio.h>
int main()
{
    int arr[100],i,search,n;
    printf("Enter the number of elements: \n");
    scanf("%d",&n);

    printf("Enter %d integers: \n",n);
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    printf("Element to be searched: \n");
    scanf("%d",&search);
    for(i=0;i<n;i++)
    {
        if(arr[i]==search)
        {
            printf("Element %d is located at %d",search,i+1);
            break;
        }
    }
    if(i==n)

```

```

    {
        printf("Element %d is not available",search);
    }

    return 0;
}

//Program 17: WAP to search an element using binary search
#include<stdio.h>
#define SIZE 10
int main()
{
    int arr[SIZE],i,item;
    printf("Enter the elements in sorted order: ");
    for(i=0;i<SIZE;i++)
        scanf("%d",&arr[i]);
    printf("Enter the element to be searched: ");
    scanf("%d",&item);
    int low=0,up=SIZE-1,mid;
    while(low<=up&&item!=arr[mid])
    {
        mid=(low+up)/2;
        if(item>arr[mid])
            mid=mid+1;
        if(item<arr[mid])
            mid=mid-1;
        if(item==arr[mid])
            printf("%d element is found at index %d ",item,mid+1);
        if(low>up)
            printf("%d element is found.",item);
    }
    return 0;
}

//Program 18: WAP to pass array elements to a function
#include<stdio.h>
int main()

{
    int arr[10],i;
    printf("Enter the array elements: ");
    for(i=0;i<10;i++){
        scanf("%d",&arr[i]);
        check(arr[i]) ;
    }
    return 0;
}

```

```

}
void check(int num){
    if(num%2==0)
        printf("%d is Even\n",num);
    else
        printf("%d is odd\n",num);
}

```

**//Program 19: WAP to pass the Array as an Function Argument**

```

#include<stdio.h>
int main()
{
    int i,arr[6]={1,2,3,4,5,6};
    func(arr);
    printf("Contents of the array are: ");
    for(i=0;i<6;i++)
        printf("%d ",arr[i]);
    printf("\n");
    return 0;
}
void func(int val[]){
    int i,sum=0;
    for(i=0;i<6;i++){
        val[i]=val[i]*val[i];
        sum=sum+val[i];
    }
    printf("Sum of Squares =%d\n",sum);
}

```

**//Program 20: WAP to use general function that works on arrays of different Functions**

```

#include<stdio.h>
int sum(int,int);
int main ( )
{
    int a[5]={2,4,6,8,10};
    int b [8] = {1, 3,5,7,9,11,13,15} ;
    int c [10] = {1, 2,3,4,5,6,7,8,9,10} ;
    printf("Sum of elements of array a are %d \n",add(a,5));
    printf("Sum of elements of array b are %d \n",add(b,8));
    printf("Sum of elements of array c are %d \n",add(c,10));

    return 0;
}
int add(int arr[],int n){
    int sum=0,i;

```



```

    for(i=0;i<n;i++)
        sum+=arr[i];
    return sum;
}
//Program 21: WAP to input and display a matrix
#include<stdio.h>
#define ROW 3
#define COL 4
int main()
{
    int mat[ROW][COL],i,j;
    printf("Enter the elements of matrix (%d x %d) row-wise\n",ROW,COL);
    for(i=0;i<ROW;i++)
        for(j=0;j<COL;j++)
            scanf("%d",&mat[i][j]);
    printf("The matrix that you have entered is : \n");
    for(i=0;i<ROW;i++)
    {
        for(j=0;j<COL;j++)
            printf("%d ",mat[i][j]);
        printf("\n");
    }

    return 0;
}

```

```

//Program 22: WAP for addition of two matrix
#include<stdio.h>
#define ROW 3
#define COL 4
int main()
{
    int mat1[ROW][COL],mat2[ROW][COL],mat3[ROW][COL],i,j;
    printf("Enter the matrix mat1(%d x %d) row-wise: \n",ROW,COL);
    for(i=0;i<ROW;i++)
        for(j=0;j<COL;j++)
            scanf("%d",&mat1[i][j]);
    printf("Enter the matrix mat2(%d x %d) row-wise: \n",ROW,COL);
    for(i=0;i<ROW;i++)
        for(j=0;j<COL;j++)
            scanf("%d",&mat2[i][j]);
    //Addition
    for(i=0;i<ROW;i++)
        for(j=0;j<COL;j++)

```

```

    mat3[i][j]=mat1[i][j]+mat2[i][j];

//Display
printf("The resultant matrix is \n");
for(i=0;i<ROW;i++)
{
    for(j=0;j<COL;j++)
        printf("%d ",mat3[i][j]);
    printf("\n");
}

return 0;
}

```

### **//Program 23: WAP to multiply two matrices**

```

#include<stdio.h>
int main(){
    int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
    printf("enter the number of row=");
    scanf("%d",&r);
    printf("enter the number of column=");
    scanf("%d",&c);
    printf("enter the first matrix element=\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            scanf("%d",&a[i][j]);
        } }
    printf("enter the second matrix element=\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++) {
            scanf("%d",&b[i][j]);
        } }

    printf("multiply of the matrix=\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            mul[i][j]=0;
            for(k=0;k<c;k++){
                mul[i][j]+=a[i][k]*b[k][j];
            } } }
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            printf("%d\t",mul[i][j]);

```

```

    }
    printf("\n");
}
return 0;
}
//Program 24: WAP to Transpose a Matrix
#include<stdio.h>
int main() {
    int mat[3][4],trans[4][3],i,j;
    printf("Enter the elements in 3* 4: \n");
    for(i=0;i<3;i++){
        for(j=0;j<4;j++){
            scanf("%d",&mat[i][j]);
        }
    }
    //Transpose
    for(i=0;i<4;i++){
        for(j=0;j<3;j++){
            trans[i][j]=mat[j][i];
        }
    }
    //display
    for(i=0;i<4;i++){
        for(j=0;j<3;j++){
            printf("%d ",trans[i][j]);
            printf("\n");
        }
    }
    return 0;
}

```

### Problem Exercise (Arrays)

1. Write a program to accept n numbers and display the sum of the highest and lowest numbers.
2. Write a program to accept n numbers in an array and display the addition of all even numbers and multiplication of all odd numbers.
3. Write a program to sort numbers of a one-d array in descending order using (i) selection sort (ii) bubble sort (iii) insertion sort
4. Write a function to reverse only first n elements of an array.
5. Write a program to modify the elements of an array such that the last element becomes the first element of the array and all other elements are shifted to the right. 1 2 3 4 5 6 7 8 9 - 9 1 2 3 4 5 6 7 8  
We can say that we have rotated the array to the right by one element. Now modify the 'above program so that we can rotate the array by any number of elements. For example when we rotate the array by 3 elements the result would be 1 2 3 4 5 6 7 8 9 - 7 8 9 1 2 3 4 5 6
6. Write a program to find out the determinant of a matrix.
7. Write a program to count the occurrences of a number in a matrix.

8. Write a program to store all the elements of a 2-D array in a 1-D array row-wise.
9. Write a program to find out whether a matrix is symmetric or not. A matrix is symmetric if the transpose of the matrix is equal to the matrix.
10. Write a program to check that the elements of an array are distinct.
11. Write a program to check that the elements of a matrix are distinct.
12. Write a program to find out the sum of elements of principal and secondary diagonals of a square matrix.
13. Write a program to enter a square matrix of odd size and then check whether it is a magic matrix or not. Any matrix is a magic matrix if all the elements in it are distinct and the sum of elements in each row, column and diagonal are equal.
14. Write a program to print the elements of a matrix spirally. For example if the matrix is

258  
1 3 7

629 The output should be 2 5 8 7 9 2 6 1 3

15. Write a program to reverse the rows of a matrix.

1 2 3 4	13 14 15 16
5 6 7 8	9 10 11 12
9 10 11 12	5 6 7 8
13 14 15 16	1 2 3 4

16. Write a program to reverse the columns of a matrix

1 2 3 4	4 3 2 1
5 6 7 8	8 7 6 5
9 10 11 12	12 11 10 9
13 14 15 16	16 15 14 13

## Pointers

**Program 1 //WAP to print the address and value of a variable**

```
#include<stdio.h>
int main()
{
    int age=12;
    float sal=123.00;
    printf("Value of age= %d and Address of age = %u\n",age,&age);
    printf("Value of Sal= %f and Address of sal = %u\n",sal,&sal);

    return 0;
}
```

**Program 2//WAP to dereference pointer variables**

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

```

{
    int a=78; //1000
    float b=4.5;
    int *p1=&a;
    float *p2=&b;

    printf("Value of p1 = Address of &a = %u\n",p1);
    printf("Value of p2 = Address of &b = %u\n",p2);
    printf("Address of p1 = %u\n",&p1);
    printf("Address of p2 = %u\n",&p2);
    printf("value of a = %d %d %d\n",a,*p1,*(&a));
    printf("value of b = %f %f %f\n",b,*p2,*(&b));
    return 0;
}

```

### **Program 3//WAP to show pointer arithmetic**

```

#include<stdio.h>
int main()
{
    int a=3, *pi=&a;
    char b='g', *pc=&b;
    float c=3.45, *pf=&c;

    printf("Value of pi  Address of a = %u\n",pi);
    printf("Value of pc  Address of b = %u\n",pc);
    printf("Value of pf  Address of c = %u\n",pf);
    pi++;
    pc++;
    pf++;
    printf("Now Value of pi  %u\n",pi);
    printf("Now Value of pc  %u\n",pc);
    printf("Now Value of pf  %u\n",pf);
    return 0;
}

```

### **Program 4 //WAP to understand pointer to pointer**

```

#include<stdio.h>
int main()
{
    int a=5;
    int *pa;
    int **ppa;
    pa=&a;
    ppa=&pa;
    printf("Address of a =%u\n",&a);
    printf("Value of pa = Address of a =%u\n",pa);
}

```

```

printf("Value of *pa = Value of a =%d\n",*pa);
printf("Address of pa =%u\n",&pa);
printf("Value of *ppa = value of pa %u\n",*ppa);
printf("Value of **ppa = Value of a %d\n",**ppa);

return 0;
}

```

**//Program 5: WAP to print the value and address of the elements of an array**

```

#include<stdio.h>
int main()
{
    int arr[5]={1,2,3,4,5};
    int i;
    for(i=0;i<5;i++)
    {
        printf("value of arr[%d] =%d\t",i,arr[i]);
        printf("Address of arr[%d] =%u\n",i,&arr[i]);
    }

    return 0;
}

```

**//Program 6:WAP to print the value and address of the elements of an array using pointer notation**

```

#include<stdio.h>
int main()
{
    int arr[5]={1,2,3,4,5};
    int i;
    for(i=0;i<5;i++)
    {
        printf("Value of arr[%d] = %d\t",i,*(arr+i));
        printf("Address of arr[%d] = %u\n",i,arr+i);
    }

    return 0;
}

```

**//Program 7: WAP to print the value of array elements using pointer and subscript notation.**

```

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

    int arr[5]={5,10,15,20,25};
    int i=0;
    for(i=0; i<5; i++)

```

```

{
    printf ("Value of arr[%d] =", i) ;
    printf ("%d\t",arr[i]);
    printf ("%d\t",*(arr+i));
    printf ("%d\t", * (i+arr));
    printf ("%d\n", i[arr]) ;
    printf ("Address of arr [%d] = %u\n",i,&arr[i]) ;
}
return 0;
}

```

**//Program 8: WAP to print the value and address of array elements by subscripting a pointer variable**

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

```
int main( )
```

```
{
```

```
    int arr[5]={5,10,15,20,25};
```

```
    int i,*p;
```

```
    p=arr;
```

```
    for(i=0;i<5;i++)
```

```
    {
```

```
        printf("Address of arr[%d]= %u %u %u %u \n",i,&arr[i],arr+i,p+i,&p[i]);
```

```
        printf("Value of arr[%d]= %d %d %d %d \n",i,arr[i],*(arr+i),*(p+i),p[i]) ;
```

```
    }
```

```
    return 0;
```

```
}
```

**//Program 9: WAP to understand the pointer to an integer and pointer to an array of integers**

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

```
int main( )
```

```
{
```

```
    int *p;
```

```
    int (*ptr)[5];
```

```
    int arr[5];
```

```
    p=arr;
```

```
    ptr=arr;
```

```
    printf("p = %u, ptr = %u \n",p,ptr);
```

```
    p++;
```

```
    ptr++;
```

```
    printf("p = %u, ptr = %u \n",p,ptr);
```

```
return 0;
```

```
}
```

**//Program 10: WAP to dereference a pointer to an array**

```
#include<stdio.h>
int main ( )
{
    int arr[5]= {3, 5,6,7,9};
    int *p=arr;
    int (*ptr)[5]=arr;
    printf("p = %u, ptr = %u\n",p,ptr);
    printf("*p = %d, *ptr = %u\n",*p,*ptr);
    printf("sizeof(p) = %u, sizeof(*p) = %u\n",sizeof(p),sizeof(*p));
    printf("sizeof(ptr) = %u, sizeof(*ptr) = %u\n",sizeof(ptr),sizeof(*ptr)) ;

    return 0;
}
```

**//Program 11:WAP to print the values and address of elements of a 2D array**

```
#include<stdio.h>
int main ( )
{
    int arr [3][4] = {
        {10,11, 12, 13},
        {20, 21, 22, 23},
        {30, 31, 32, 33}
    };

    int i,j;
    for(i=0;i<3;i++){

        printf("Address of %dth array = %u %u \n",i,arr[i],*(arr+i));
        for (j= 0;j<4;j++)
            printf("%d %d ",arr[i][j],*(*(arr+i)+j));
        printf("\n");
    }
    return 0;
}
```

**//Program 12:WAP to explain call by value**

```
#include<stdio.h>
int main()
{
    int a=3,b=5;
    printf("Before calling the functions a = %d and b = %d\n",a,b);
    value(a,b);
    printf("After calling the functions a = %d and b = %d\n",a,b); //a=4 b=6 a=3 b=5

    return 0;
}
```



```

}
void value(int x,int y)
{
    x++;
    y++;
    printf("Value of x = %d and y = %d\n",x,y);
}

```

**//Program 13:WAP to explain call by reference**

```

#include<stdio.h>
int main()
{
    int a=3,b=5;//a=1000 b=2000
    printf("Before calling the functions a = %d and b = %d\n",a,b);
    value(&a,&b);
    printf("After calling the functions a = %d and b = %d\n",a,b); //a=4 b=6

    return 0;
}
void value(int *x,int *y)
{
    (*x)++;
    (*y)++;
}

```

**//Program 14: WAP to show how to return more than one value from a function using call by reference**

```

#include<stdio.h>
int main()
{
    int a=5,b=3,sum,diff,prod;
    func(a,b,&sum,&diff,&prod);
    printf("Sum = %d, Diff =%d, Prod = %d ",sum,diff,prod);

    return 0;
}
void func(int x,int y,int *ps,int *pd,int *pp)
{
    *ps=x+y;
    *pd=x-y;
    *pp=x*y;
}

```

**//WAP to show the Dynamic Allocation of memory using malloc()**

```

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

```

```

int main( )
{
    int *p,n,i;
    printf("Enter the number of integers to be entered ") ;
    scanf("%d" ,&n) ;
    p=(int *)malloc(n*sizeof(int));
    if (p==NULL)
    {
        printf ("Memory not available\n");
        exit(1);
    }
    for(i=0;i<n;i++){
        printf ("Enter an integer ");
        scanf("%d",p+i);
    }
    for(i=0;i<n;i++)
        printf("%d\t",*(p+i));
return 0;
}
//WAP to show the Dynamic Allocation of memory using calloc ( )
#include<stdio.h>
#include<stdlib.h>
int main( )
{
    int *p,n,i;
    printf("Enter the number of integers to be entered ") ;
    scanf("%d" ,&n) ;
    p=(int *)calloc(n,sizeof(int));
    if (p==NULL)
    {
        printf ("Memory not available\n");
        exit(1);
    }
    for(i=0;i<n;i++){
        printf ("Enter an integer ");
        scanf("%d",p+i);
    }
    for(i=0;i<n;i++)
        printf("%d\t",*(p+i));
return 0;
}

//WAP to show the Dynamic Allocation of memory using realloc ( )
#include<stdio.h>

```

```

#include<stdlib.h>
int main( )
{
    int *p,n,i;
    printf("Enter the number of integers to be entered(5) ");
    scanf("%d",&n);
    p=(int *)calloc(n,sizeof(int));
    if (p==NULL)
    {
        printf ("Memory not available\n");
        exit(1);
    }
    for(i=0;i<n;i++){
        printf ("Enter an integer ");
        scanf("%d",p+i);
    }
    p= (int *)realloc(p,9*sizeof (int) );
    if (p==NULL)
    {
        printf ("Memory not available\n");
        exit(1);
    }
    printf("Enter 4 more integer: \n");
    for(i=5;i<9;i++){
        printf ("Enter an integer ");
        scanf("%d",p+i);
    }
    for(i=0;i<9;i++)
        printf("%d\t",*(p+i));
    return 0;
}

```

## Strings

**Program 1://WAP to initialize String Variables and print the characters and address**

```

#include<stdio.h>
int main()
{
    //char str[]={ 'F','a','i','s','a','l','\0'};
    char str[]="Faisal";
    for(int i=0;str[i]!='\0';i++)
    {

```

```

        printf("Character = %c\t", str[i]);
        printf("Address = %p\n", &str[i]);

    }
    return 0;
}

```

**Program 2: //WAP to print the address and characters of a string using pointers**

```

#include<stdio.h>
int main()
{
    char str[]="Faisal";
    char *p;
    p=str;
    while(*p!='\0')
    {
        printf("Character = %c\t", *p);
        printf("Address = %p\n", p);
        p++;
    }
    return 0;
}

```

**Program 3: //WAP to input and output using scanf() and printf()**

```

#include<stdio.h>
int main()
{
    char name[30];
    printf("Enter a name: ");
    scanf("%s", name);
    printf("Name= %s", name); //Faisal\0' Firdous
    return 0;
}

```

**Program 4: //WAP to input and output using gets() and puts()**

```

#include<stdio.h>
int main()
{
    char name[30];
    printf("Enter a name: ");
    gets(name);
    puts(name);

    return 0;
}

```

**Program 5://WAP to understand the strlen() function**

```

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

    char str[30];
    int len;
    printf("Enter the string: ");
    gets(str);
    len=strlen(str);
    printf("Length of a string is : %d",len);

    return 0;
}

```

**Program 6: //WAP to understand the strlen() function using user defined function**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[30];
    int len;
    printf("Enter the string: ");
    gets(str);
    printf("Length of a string is %d",astrlen(str));
    return 0;
}
int astrlen(char str[]){
    int i=0;
    while(str[i]!='\0'){
        i++;
    }
    return i;
}

```

**//WAP to understand the strlen() function using pointer version**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[30];
    int len;
    printf("Enter the string: ");

```

```

    gets(str);
    printf("Length of a string is %d",astrlen(str));
    return 0;
}
int astrlen(char *str)
{

    char *sstr=str;
    while(*str!='\0')
    {
        str++;
    }
    return str-sstr;
}

```

**Program 7: //WAP to understand the strcmp() function**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    *if((strcmp(str1,str2))==0)
        printf("Strings are same");
    else
        printf("Strings are not same");
    return 0;
}

```

**Program 8: //WAP to understand the strcmp() function using user-defined function**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    if(astrcmp(str1,str2))

```

```

        printf("Strings are not same");
    else
        printf("Strings are same");

    return 0;
}
int strcmp(char str1[],char str2 []){
    int i=0;
    while(str1[i]!='\0'&&str2[i]!='\0'&&str1[i]==str2[i])
        i++;
    if(str1[i]==str2[i])
        return 0;
    else
        return(str1[i]-str2[i]);

}
//WAP to understand the strcmp( ) function using pointer version
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    if(strcmp(str1,str2))
        printf("Strings are not same");
    else
        printf("Strings are same");

    return 0;
}
int strcmp(char *str1,char *str2 ){

while(*str1!='\0'&&*str2!='\0'&&*str1==*str2)
    str1++;
    str2++;
if(*str1==*str2)
    return 0;
else
    return(*str1-*str2);

}

```

**Program 9: //WAP to understand the strcpy() function**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];

    printf("Enter the second string: ");
    scanf("%s",str2);
    printf("First string: %s \t Second string : %s\n",str1,str2);
    strcpy(str1,"India");
    strcpy(str2,"China");
    printf("First string: %s \t Second string : %s",str1,str2);
    return 0;
}
```

**Program 10: //WAP to understand the strcpy() function using user-defined function**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the second string: ");
    scanf("%s",str2);
    strcpy(str1,str2);

    return 0;
}

void strcpy(char str1[],char str2[]){
    int i=0;
    while(str2[i]!='\0'){
        str1[i]=str2[i];
        i++;
    }
    str1[i]='\0';
    printf("str1 = %s",str1);
}
```

**//Program 11: WAP to understand the strcpy() function using pointer version**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the second string: ");
```



```

scanf("%s",str2);
strcpy(str1,str2);

printf("str1 = %s",str1);
return 0;
}
void strcpy(char *str1,char *str2){
int i=0;
while(*str2!='\0'){
    *str1=*str2;
    str1++;
    str2++;
}
*str1='\0';
}

```

**Program 12://WAP to understand the strcat() function**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    printf("First string : %s \t Second string : %s\n",str1,str2);
    strcat(str1,"_Banana");
    printf("Now first string is %s",str1);
    return 0;
}

```

**Program 13://WAP to understand the strcat() function using user-defined function**

```

#include<stdio.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    astrcat(str1,str2);
    return 0;
}

```

```

void astrcat(char str1[],char str2[])
{
    int i=0,j=0;//apple
    while(str1[i]!='\0'){
        i++;
    }
    while(str2[j]!='\0'){
        str1[i]=str2[j];
        i++;
        j++;
    }
    str1[i]='\0';
    printf("After concat string = %s",str1);
}

```

**//Program 14:WAP to understand the strcat() function using pointer version**

```

#include<stdio.h>
int main()
{
    char str1[20],str2[20];
    printf("Enter the first string: ");
    scanf("%s",str1);
    printf("Enter the second string: ");
    scanf("%s",str2);
    astrcat(str1,str2);
    printf("After concat string = %s",str1);
    return 0;
}

void astrcat(char *str1,char *str2)
{
    while(*str1!='\0'){
        str1++;
    }
    while(*str2!='\0'){
        *str1=*str2;
        str1++;
        str2++;
    }
    *str1='\0';
}

```

**Program 15://WAP to test whether the word is Palindrome or Not.**

```

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

```

```

int main()
{
    char str[20]; //abcdcbad
    int j,i=0,flag;
    printf("Enter the word: ");
    scanf("%s",str);
    j=strlen(str)-1;
    while(i<=j){
        if(str[i]==str[j])
            flag=1;

        else{
            flag=0;
            break;
        }
        i++;
        j--;
    }
    if(flag==1)
        printf("Word is pallindrome.");
    else
        printf("Word if not pallindrome");
    return 0;
}

```

**Program 16://WAP to convert lowercase into uppercase**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[10];
    int i=0;
    printf("Enter a string in lowercase: ");
    scanf("%s",str); //apple
    while(str[i]!='\0'){
        str[i]=str[i]-32;
        i++;
    }
    printf("The uppercase string is : %s",str);
    return 0;
}

```

**Program 17://WAP to accept any string and count number of words in it**

```

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

```

```

int main()
{
    char line[100];
    int count=0,i=0;
    printf("Enter the line of text: ");
    gets(line); //apple banana cherry
    while(line[i]!='\0'){
        if(line[i]==32)
            count++;
        i++;
    }
    if(line[i]=='\0')
        count++;
    printf("The number of words in line =%d",count);
    return 0;
}

```

**Program 18: //WAP to enter any string and print its reverse order**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[10];
    int len;
    printf("Enter any string: ");
    scanf("%s",str);
    len=strlen(str)-1;
    while(len>=0){
        printf("%c",str[len]);
        len--;
    }

    return 0;
}

```

**//Program 19: WAP to print the strings of two dimensional character array**

```

#include<stdio.h>
#include<string.h>
#define N 5
#define LEN 10
int main()
{
    char arr[N][LEN]={"white",
        "red",
        "blue",

```

```

        "green",
        "brown"
    };
    int i;
    for(i=0;i<N;i++)
    {
        printf("str =%s\t",arr[i]);
        printf("Address of str =%u\n",arr[i]);

    }
    return 0;
}

```

**//Program 20: WAP to check whether the word is palindrome or not**

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[10];
    int i=0,j,flag;
    printf ("Enter the word ") ;
    scanf("%s",str) ;
    j=strlen(str)-1;
    while (i<=j)
    {
        if(str[i]==str[j])
            flag=1;
        else
            flag=0;
        break;
        i++;
        j--;
    }
    if(flag==1)
        printf ("Word is palindrome\n");
    else
        printf ("Word is not palindrome\n");

    return 0;
}

```

**//Program 21: WAP to find the substring from string**

```

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

```

```

int main() {
    char str[100], substr[50];
    int i, j, found = 0, strLen, substrLen;

    printf("Enter the main string: ");
    scanf("%s",str);

    printf("Enter the substring: ");
    scanf("%s",substr);

    strLen = strlen(str);
    substrLen = strlen(substr);
    for(i = 0; i <= strLen - substrLen; i++) {
        for(j = 0; j < substrLen; j++) {
            if(str[i + j] != substr[j]) {
                break;
            }
        }
        if(j == substrLen) { // Substring found
            found = 1;
            printf("Substring found at position: %d\n", i + 1);
            break;
        }
    }

    if(!found) {
        printf("Substring not found.\n");
    }

    return 0;
}

```

## Programming Exercise

1. Write a function for performing case insensitive string comparison
2. Write a program to accept a line of text and display the number of consonants and spaces in that line of text.
3. Write a function that searches for a character in the string and returns the number of occurrences of that character in the string. It should take two arguments, first a string and then a character.
4. Write a function which replaces all the occurrences of a character from a string with another character. It should take three arguments, a string and two characters.
5. Write a function which deletes all the occurrences of a character from a string. It should take two arguments, a string and a character.

6. Write a program to accept a line of text and a word. Display the number of occurrences of that word in the text.
7. Write a function to remove all the leading and trailing blanks from a string.
8. Write a program to input text and replace all the occurrences of word "Calcutta" by "Kolkata" in that text.
9. Write a program to accept a line of text and print that text after removing all spaces and delimiters.
10. Write a program to accept any 10 names and display those names after sorting them alphabetically in descending order.
11. Write a program to encode text and to decode the encoded text.
  - (i) Perform the encoding so that every character is replaced by its next character. For example, replace a by b, b by c and so on. Replace z by a.  
 Plain text: program  
 Encoded text : qsphsbn  
 Decoded text : program
  - (ii) Perform the encoding according to these replacements a b c d e f g h i j k l m n o p r s t u v w x y z.  
 Plain text: program  
 Encoded text :cjqtjmp  
 Decoded text :program
12. Input a string and change it so that the characters are placed in alphabetical order. For example the string "motivate" should be changed to "aeimotv"
13. Write a program to abbreviate input text. For example if the input is "World Health Organization", then the output should be WHO.
14. Write a function to extract a substring from a string. Assume that the substring starts at the i<sup>th</sup> character and is n characters long.
15. Write a program to input a number from 0 to 6 and print the corresponding day using an array of pointers. For example if 0 is entered then print Sunday, if 1 is entered print Monday.

## Structure

### **Program 1: //WAP to display the values of structure members**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    float marks;
};
int main()
{
    struct student stu1={"abc",234,89.67};
    printf("Name =%s\n",stu1.name);
    printf("Age =%d\n",stu1.rollno);
    printf("Marks =%f",stu1.marks);
    return 0;
}
```

```
}
```

**Program 2: //WAP to ask user enter the details and display the values of structure members**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    float marks;
};
int main()
{
    struct student stu1;
    printf("Enter your name: ");
    scanf("%s",stu1.name);
    printf("Enter your roll no: ");
    scanf("%d",&stu1.rollno);
    printf("Enter your marks: ");
    scanf("%f",&stu1.marks);

    printf("Name =%s\n",stu1.name);
    printf("Age =%d\n",stu1.rollno);
    printf("Marks =%f",stu1.marks);
    return 0;
}
```

**Program 3: //WAP to ask user enter the details and display the address of members**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    float marks;
};
int main()
{
    struct student stu1;
    printf("Enter your name: ");
    scanf("%s",stu1.name);
    printf("Enter your roll no: ");
    scanf("%d",&stu1.rollno);
    printf("Enter your marks: ");
    scanf("%f",&stu1.marks);

    printf("Address of Name =%u\n",stu1.name);
    printf("Address of Age =%u\n",&stu1.rollno);
    printf("Address of Marks =%u",&stu1.marks);
    return 0;
}
```



```
}
```

**Program 4://WAP to ask user enter the details and check the size of structure**

```
#include<stdio.h>
struct student{
    /*char name[20];
    int rollno;
    float marks;*/
    char ch;
    int num;
};
int main()
{
    struct student stu1;
    printf("Size of struct =%d",sizeof(stu1));
    /*printf("Enter your name: ");
    scanf("%s",stu1.name);
    printf("Enter your roll no: ");
    scanf("%d",&stu1.rollno);
    printf("Enter your marks: ");
    scanf("%f",&stu1.marks);

    printf("Address of Name =%u\n",stu1.name);
    printf("Address of Age =%u\n",&stu1.rollno);
    printf("Address of Marks =%u",&stu1.marks);*/
    return 0;
}
```

**Program 5://WAP to understand array of structures**

```
#include<stdio.h>
struct student{
    char name[30];
    int rollno;
    float marks;
};
int main()
{
    struct student s1[10];
    int i;
    for(i=0;i<5;i++){
        printf("Enter name, rollno and marks of student %d: ",i+1);

        scanf("%s%d%f",s1[i].name,&s1[i].rollno,&s1[i].marks);
    }
}
```

```

    for(i=0;i<5;i++){
        printf("Name = %s \t RollNo = %d\t Marks =%f\n",s1[i].name,s1[i].rollno,s1[i].marks);
    }
    return 0;
}

```

**Program 6: //WAP to understand arrays within structures**

```

#include<stdio.h>
struct student{
    char name [20] ;
    int rollno;
    int submarks[4] ;
};
int main()
{
    int i,j;
    struct student stuarr [3] ;
    for(i=0;i<3;i++){
        printf ("Enter data for student %d\n", i+1);
        printf ("Enter name ") ;
        scanf("%s",stuarr[i].name);
        printf ("Enter roll number ") ;
        scanf(" %d",&stuarr[i].rollno) ;

        for(j=0;j<4;j++)
        {
            printf ("Enter marks for subject %d: ",j+1);
            scanf("%d",&stuarr[i].submarks[j]);
        }
    }
    for(i=0;i<3;i++)
    {
        printf("Data of student %d\n", i+1);
        printf("Name %s, Roll number %d\n Marks =%f ",stuarr[i].name,stuarr[i].rollno);
        for(j=0;j<4;j++)
            printf("%d",stuarr[i].submarks[j]);
        printf ("\n");
    }
    return 0;
}

```

**Program 7//WAP to understand the concept of Structure within Structure**

```

#include<stdio.h>
struct date{
    int day;

```

```

        int month;
        int year;
    }d1;
struct student
{
    char name[20];
    int rollno;
    struct date d1;
    float marks;

} s1;
struct employee
{
    char name[20];
    int empid;
    struct date d1;

} e1;

int main()
{
    s1.d1.day=2;

    return 0;
}

```

**Program 8: //WAP to understand the concept of Pointer to Structure**

```

#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    int marks;

};
int main()
{
    struct student s1={"abc",123,45};
    struct student *p=&s1;
    printf("Name = %s\n",p->name);
    printf("RollNo = %d\n",p->rollno);
    printf("Marks = %d\n",p->marks);

}

```

**Program 9: //WAP to understand the concept of Pointer within Structure**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    int marks;
    int *ptr;

};
int main()
{
    struct student s1,*p;
    *p->ptr;

}
```

**Program 10: //WAP to understand the concept structure and Functions(Passing Member as Arguments)**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
    int marks;
};
void display(char name[],int rollno,int marks);
int main()
{
    struct student s1={"abc",1,45};
    display(s1.name,s1.rollno,s1.marks);
    return 0;

}
void display(char name[],int rollno,int marks){
    printf("Name = %s\n",name);
    printf("Rollno = %d\n",rollno);
    printf("Marks = %d\n",marks);

}
```

**Program 11: //WAP to understand the concept structure and Functions(Passing structure variable)**

```
#include<stdio.h>
struct student{
    char name[20];
    int rollno;
```

```

    int marks;
};
void display(struct student);
int main()
{
    struct student s1={"abc",12,234};

    display(s1);
    return 0;

}
void display(struct student s1)
{
    printf("Name = %s\n",s1.name);
    printf("RolNo = %d\n",s1.rollno);
    printf("marks = %d\n",s1.marks);
}

```

**Program 12//WAP to pass Pointers to Structure as Arguments.**

```

#include<stdio.h>
struct student{
    char name[10];
    int rollno;
    int marks;

};
void inc_marks(struct student *stu);
void display(struct student *s);
int main()
{
    struct student s1={"abc",34,89};
    struct student s2={"def",37,80};
    inc_marks(&s1);
    inc_marks(&s2);
    display(&s1);
    display(&s2);

    return 0;
}
void inc_marks(struct student *stu)
{
    (stu->marks)++;
}
void display(struct student *s)
{

```

```

printf("Name =%s\n",s->name);
printf("RollNo =%d\n",s->rollno);
printf("Marks =%d\n",s->marks);

}

```

### **Program 13//WAP to show the returning a Structure variable from function**

```

#include<stdio.h>
struct student
{
    char name[10];
    int rollno;
    int marks;
};
struct student change(struct student s);
void display(struct student stu);
int main()
{
    struct student s1={"mary",45,87};
    struct student s2={"hary",48,68};
    s1=change(s1);
    s2=change(s2);
    display(s1);
    display(s2);

    return 0;
}
struct student change(struct student s)
{
    s.marks=s.marks+15;
    s.rollno=s.rollno+2;
    return s;
};
void display(struct student stu)
{
    printf("Name %s\n",stu.name);
    printf("RollNo %d\n",stu.rollno);
    printf("Marks %d\n",stu.marks);
}

```

### **Program 14//WAP to show Returning a Pointer to Structure From a Function**

```

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

```

```

#include<string.h>
struct student
{
    char name[10];
    int rollno;
    int marks;
};
struct student *func();
void display(struct student *);
int main()
{
    struct student *sptr;
    sptr=func();
    display(sptr);

    return 0;
}
struct student *func()
{
    struct student *ptr;
    ptr=(struct student *)malloc(sizeof(struct student));
    strcpy(ptr->name,"Jenny");
    ptr->rollno=56;
    ptr->marks=79;
    return ptr;
};
void display(struct student *stu)
{
    printf("Name =%s\n",stu->name);
    printf("RollNo =%d\n",stu->rollno);
    printf("Marks =%d\n",stu->marks);
}

```

### **Program 15//WAP to Pass Array of Structures as Arguments**

```

#include<stdio.h>
struct student
{
    char name[10];
    int rollno;
    int marks;
};
void updateMarks(struct student sarr[]);
void display(struct student stu);
int main()

```

```

{
    struct student sarr[3]={{"abc",12,89},
                            {"def",13,79},
                            {"ghi",14,68}};
    updateMarks(sarr);
    display(sarr[0]);
    display(sarr[1]);
    display(sarr[2]);

    return 0;
}
void updateMarks(struct student sarr[])
{
    int i;
    for(i=0;i<3;i++)
    {
        sarr[i].marks=sarr[i].marks+10;
        sarr[i].rollno=sarr[i].rollno+1;
    }
}
void display(struct student stu)
{
    printf("Name =%s\t",stu.name);
    printf("RollNo =%d\t",stu.rollno);
    printf("Marks =%d\n",stu.marks);
}

```

**Program 16 //WAP for accessing union members**

```

#include<stdio.h>
union Student
{
    int marks;
    char grade;
    float per;
};
int main()
{
    union Student s1;
    s1.marks=89;
    printf("Marks =%d\n",s1.marks);
    s1.grade='A';
    printf("Grade =%c\n",s1.grade);
    s1.per=89.5;
}

```



```

    printf("Percentage =%f\n",s1.per);

    return 0;
}

```

**Program 17 //WAP to compare the memory allocated for a union and structure**

```

#include<stdio.h>
struct stag
{
    char c;
    int i;
    float f;
};
union utag
{
    char c;
    int i;
    float f;
};
int main()
{
    struct stag s;
    union utag u;
    printf("Size of stag =%u\n",sizeof(s));
    printf("Size of utag =%u\n",sizeof(u));
    printf("Address of stag =%u\n",&s);
    printf("Address of utag =%u\n",&u);
    printf("Address of stag members %u %u %u\n",&s.c,&s.i,&s.f);
    printf("Address of utag members %u %u %u\n",&u.c,&u.i,&u.f);

    return 0;
}

```

## Recursion

**Program 1://WAP to show the Head Recursion concept**

```

#include <stdio.h>
void fun(int n)
{
    if(n>0)
    {
        fun(n-1);
        printf("%d ",n);
    }
}

```

```

int main()
{
    int x=3;
    fun(x);
return 0;
}

```

**Program 2: //WAP to show the Tail Recursion concept**

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

```
void fun(int n)
```

```

{
if(n>0)
{
    printf("%d ",n);
    fun(n-1);
}
}

```

```
int main()
```

```

{
    int x=3;
    fun(x);
return 0;
}

```

**//Program 3: WAP to show the concept of Recursion**

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

```
int fun(int n)
```

```

{
if(n>0)
{
    return fun(n-1)+n;
}
return 0;
}

```

```
int main()
```

```

{
    int a=5;
    printf("%d ",fun(a));
return 0;
}

```

**//Program 4: WAP to show the concept of static variable in Recursion**

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

```
int fun(int n)
```

```

{
    static x=0;
if(n>0)
{
    x++;
    return fun(n-1)+x;
}
return 0;
}
int main()
{
    int a=5;
    printf("%d ",fun(a));
return 0;
}

```

**//Program 5: WAP to show the concept of global variable in Recursion**

```

#include <stdio.h>
int x=0;
int fun(int n)
{
if(n>0)
{
    x++;
    return fun(n-1)+x;
}
return 0;
}
int main()
{
    int a=5;
    printf("%d ",fun(a));
return 0;
}

```

**//Program 6: WAP to show the concept of Tree Recursion**

```

#include <stdio.h>
void fun(int n)
{
if(n>0)
{
    printf("%d ",n);
    fun(n-1);
    fun(n-1);
}
}

```

```

}
}
int main()
{
    fun(3);
    return 0;
}

```

**//Program 7: WAP to show the concept of Indirect Recursion**

```

#include <stdio.h>
void funB(int n);
void funA(int n)
{
    if(n>0)
    {
        printf("%d ",n);
        funB(n-1);
    }
}
void funB(int n)
{
    if(n>1)
    {
        printf("%d ",n);
        funA(n/2);
    }
}
int main()
{
    funA(20);
    return 0;
}

```

**//Program 8: WAP to show the concept of Nested Recursion**

```

#include <stdio.h>
int fun(int n)
{
    if(n>100)
        return n-10;
    return fun(fun(n+11));
}
int main()
{
    int r;
    r=fun(95);
    printf("%d\n",r);
}

```

```

return 0;
}
//Program 9:WAP to show the Sum of N natural numbers

```

```

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

```

```

int main()
{
    int r=sum(5);
    printf("%d ",r);
    return 0;
}

```

```

//Program 10:WAP to show the concept for finding factorial of n

```

```

#include<stdio.h>
int fact(int n)
{
    if(n==0)
        return 1;
    return fact(n-1)*n;
}

```

```

int main()
{
    int r=fact(5);
    printf("%d ",r);
    return 0;
}

```

```

//Program 11 Method 1:WAP to show the concept for finding Power ( $m^n$ )

```

```

#include<stdio.h>
int power(int m,int n)
{
    if(n==0)
        return 1;
    return pow(m,n-1)*m;
}

```

```

int main()
{

```

```

    int r=power(2,9);
    printf("%d ",r);
    return 0;
}

//Program 12 Method 2:WAP to show the concept for finding Power ( $m^n$ )
#include<stdio.h>
int power1(int m,int n)
{
    if(n==0)
        return 1;
    if(n%2==0)
        return power(m*m,n/2);
    else
        return m*power(m*m,n/2);
}

int main()
{
    int r=power1(2,9);
    printf("%d ",r);
    return 0;
}

//Program 13 Method 1: WAP to show the concept Taylor Series using Static variables
#include<stdio.h>
double e(int x, int n)
{
    static double p=1,f=1;
    double r;
    if(n==0)
        return 1;
    r=e(x,n-1);
    p=p*x;
    f=f*n;
    return r+p/f;
}

int main()
{
    printf("%lf\n",e(4,15));
    return 0;
}

//Program 14 Method 2: WAP to show the concept Taylor Series Horner's Rule
#include <stdio.h>
double e(int x, int n)
{

```

```

static double s=1;
if(n==0)
    return s;
s=1+x*s/n;
return e(x,n-1);
}
int main()
{
    printf("%lf\n",e(2,10));
    return 0;
}

```

**//Program 15 Method 3: WAP to show the concept Taylor Series Iterative**

```

#include <stdio.h>
double e(int x, int n)
{
    double s=1;
    int i;
    double num=1;
    double den=1;
    for(i=1;i<=n;i++)
    {
        num*=x;
        den*=i;
        s+=num/den;
    }
    return s;
}
int main()
{
    printf("%lf\n",e(1,10));
    return 0;
}

```

**//Program 16 Method 1:WAP to show the concept Fibonacci series using Iterative Method**

```

#include <stdio.h>
int fib(int n)
{
    int t0=0,t1=1,s=0,i;
    if(n<=1)
        return n;
    for(i=2;i<=n;i++)
    {
        s=t0+t1;
        t0=t1;

```

```

        t1=s;
    }
    return s;
}
int main()
{
    int i;
    for(i=0;i<10;i++)
        printf("%d \t",fib(i));
    return 0;
}

```

**//Program 17 Method 2:WAP to show the concept Fibonacci series using Functions**

```

#include <stdio.h>
int rfib(int n){
    if(n<=1)
        return n;
    return rfib(n-2)+rfib(n-1);
}
int main()
{
    int i;
    for(i=0;i<10;i++)
        printf("%d \t",rfib(i));
    return 0;
}

```

**//Program 18 Method 3:WAP to show the concept Fibonacci series using Recursion (Memoization)**

```

#include <stdio.h>
int F[10];
int mfib(int n){
    if(n<=1){
        F[n]=n;
        return n;
    }
    else {
        if(F[n-2]==-1)
            F[n-2]=mfib(n-2);
        if(F[n-1]==-1)
            F[n-1]=mfib(n-1);
        F[n]=F[n-2]+F[n-1];
    }

    return F[n-2]+F[n-1];
}
}

```



```

int main()
{
    int i;
    for(i=0;i<10;i++){
        F[i]=-1;
        printf("%d \t",mfib(i));
    }
    return 0;
}

```

**//Program 19 Method 1:**WAP to show the Combination Formula using functions

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

```

int fact(int n)
{
    if(n==0)
        return 1;
    return fact(n-1)*n;
}

```

```

int nCr(int n,int r)
{
    int num,den;
    num=fact(n);
    den=fact(r)*fact(n-r);
    return num/den;
}

```

```

int main()
{
    printf("%d \n",nCr(5,3));
    return 0;
}

```

**//Program 20 Method 2:**WAP to show the Combination Formula using Recursion

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

```

int NCR(int n,int r)
{
    if(n==r || r==0)
        return 1;
    return NCR(n-1,r-1)+NCR(n-1,r);
}

```

```

int main()
{
    printf("%d \n",NCR(5,3));
    return 0;
}

```

**//Program 21: WAP to show the Tower of Hanoi Problem**

```

#include <stdio.h>
void TOH(int n,int A,int B,int C)
{
    if(n>0)
    {
        TOH(n-1,A,C,B);
        printf("(%d,%d)\n",A,C);
        TOH(n-1,B,A,C);
    }
}
int main()
{
    TOH(3,1,2,3);
    return 0;
}

```

**Program 22 WAP that prints the reverse of a positive integer using recursion.**

**Program 23 WAP to find out the Least Common Multiple and Highest Common factor of two numbers recursively**

## Sorting

**Program 1 WAP to implement Selection Sort**

```

#include <stdio.h>
#define SIZE 10
int main()
{
    int arr[SIZE] ;
    int i,j , temp;
    printf ("Enter elements of the array : \n") ;
    for(i=0;i<SIZE;i++)
        scanf("%d",&arr[i]);
    for(i=0;i<SIZE-1;i++)
        for(j=i+1;j<SIZE-1;j++){
            if(arr[i]>arr[j]){
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    printf("Sorted Array : ");
    for(i=0;i<SIZE;i++)
        printf("%d\t",arr[i]);
    return 0;
}

```

```
}
```

**Program 2 Method 1 WAP to implement Bubble Sort**

```
#include <stdio.h>
#define SIZE 10
int main()
{
    int arr[SIZE] ;
    int i,j , temp;
    printf ("Enter elements of the array : \n") ;
    for(i=0;i<SIZE;i++)
    scanf("%d",&arr[i]);
    for(i=0;i<SIZE-1;i++){
        for(j=0;j<SIZE-1-i;j++){
            if(arr[j]>arr[j+1]){
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    printf("Sorted Array : ");
    for(i=0;i<SIZE;i++)
        printf("%d\t",arr[i]);
    return 0;
}
```

**Program 3 Method 2 WAP to implement Bubble Sort using functions**

```
#include <stdio.h>
#include<stdlib.h>
void swap(int *x,int *y)
{
    int temp=*x;
    *x=*y;
    *y=temp;
}
void Bubble(int A[],int n)
{
    int i,j,flag=0;
    for(i=0;i<n-1;i++)
    {
        flag=0;
        for(j=0;j<n-i-1;j++)
        {
            if(A[j]>A[j+1])
            {
```

```

        swap(&A[j],&A[j+1]);
        flag=1;
    }
}
if(flag==0)
break;
}
}
int main()
{
int A[]={11,13,7,12,16,9,24,5,10,3},n=10,i;
Bubble(A,n);
for(i=0;i<10;i++)
printf("%d ",A[i]);
printf("\n");
return 0;
}

```

#### **Program 4 WAP to implement Insertion Sort**

```

#include <stdio.h>
void insert(int a[], int n)
{
    int i, j, temp;
    for (i = 1; i < n; i++) {
        temp = a[i];
        j = i - 1;
        while(j>=0 && temp <= a[j])
        {
            a[j+1] = a[j];
            j = j-1;
        }
        a[j+1] = temp;
    }
}
void printArr(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}
int main()
{
    int a[] = { 12, 31, 25, 8, 32, 17 };
    int n = sizeof(a) / sizeof(a[0]);
}

```

```

printf("Before sorting array elements are - \n");
printArr(a, n);
insert(a, n);
printf("\nAfter sorting array elements are - \n");
printArr(a, n);

return 0;
}

```

### **Program 5 WAP to implement Quick Sort**

```

#include <stdio.h>
#include<stdlib.h>
void swap(int *x,int *y)
{
    int temp=*x;
    *x=*y;
    *y=temp;
}
int partition(int A[],int l,int h)
{
    int pivot=A[l];
    int i=l,j=h;
do
    {
        do {i++;} while(A[i]<=pivot);
        do {j--;} while(A[j]>pivot);
    if(i<j)
        swap(&A[i],&A[j]);
    }
    while(i<j);
    swap(&A[l],&A[j]);
return j;
}
void QuickSort(int A[],int l,int h)
{
    int j;
    if(l<h)
    {
        j=partition(A,l,h);
        QuickSort(A,l,j);
        QuickSort(A,j+1,h);    }
}
int main()
{
    int A[]={11,13,7,12,16,9,24,5,10,3},n=10,i;

```

```

QuickSort(A,n);
for(i=0;i<10;i++)
printf("%d ",A[i]);
printf("\n");
return 0;
}

```

## Files

### Program 1 WAP to understand the use of fputc () function

```

#include<stdio.h>
int main()
{
    FILE *fptr;
    int ch;
    if((fptr=open("mytext.txt","w")==NULL))
    {
        printf("File does not exist");
    }
    else{
        printf ("Enter text Ctrl+z for Stop \n" );
        while((ch=getchar())!=EOF)
        fputc (ch, fptr);
    }
    fclose(fptr);

    return 0;
}

```

### Program 2 WAP to understand the use of fgetc ( )

```

#include<stdio.h>
int main ( )
{
    FILE *p;
    char ch;
    if((p=fopen("myfile.txt", "r"))==NULL)
        printf("This file doesn't exist\n");
    else
    {
        while((ch=fgetc(p))!=EOF)
        printf("%c", ch);
    }
}

```

```

    fclose(p);
return 0;
}

```

**Program 3 WAP to understand the use of putw () function**

```

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

    FILE * fptr;
    int value;
    fptr=fopen("num.dat", "wb");
    for(value=1;value<=30;value++)
        putw(value,fptr);
    fclose (fptr) ;

    return 0;
}

```

**//Program 4 WAP to understand the use of getw () function**

```

#include<stdio.h>
int main()
{
    FILE * fptr;
    int value;
    fptr=fopen("num.dat", "rb");
    while( (value=getw(fptr)) !=EOF)
        printf("%d\t",value);
    fclose (fptr) ;
    return 0;
}

```

**Program 5 Program to understand the use of fprintf ()**

```

#include<stdio.h>
int main( )
{
    FILE *fp;
    char name[10];
    int age;
    fp=fopen("rec.dat","w") ;
    printf("Enter your name and age ") ;
    scanf("%s%d",name,&age) ;
    fprintf(fp,"My name is %s and age is %d",name,age);
    fclose(fp) ;
    return 0;
}

```

**Program 6 Program to understand the use of fputs ()**

```
#include<stdio.h>
int main ( )
{
    FILE *fptr;
    char str[80];
    fptr=fopen("test.txt", "w");
    printf ( "Enter the text \n" ) ;
    printf ("To stop entering, press Ctrl+d in Unix and Ctrl+z in Dos\n");
    while (gets(str)!=NULL)
        fputs(str,fptr);
    fclose(fptr) ;
return 0;
}
```

### **Program 7 Program to understand the use of fgets ( )**

```
#include<stdio.h>
int main ( )
{
    FILE *fptr;
    char str[80];
    fptr=fopen("test.txt", "r");
    printf ( "Enter the text \n" ) ;
    printf ("To stop entering, press Ctrl+d in Unix and Ctrl+z in Dos\n");
    while(fgets(str,80,fptr)!=NULL)
        puts(str);
    fclose(fptr) ;
return 0;
}
```

### **Program 8 Program to understand the use of fprintf ( )**

```
#include<stdio.h>
int main( )
{
    FILE *fp;
    char name[10];
    int age;
    fp=fopen("rec.dat","w") ;
    printf("Enter your name and age ") ;
    scanf("%s%d",name,&age) ;
    fprintf(fp,"My name is %s and age is %d",name,age);
    fclose(fp) ;
    return 0;
}
```



**Program 9 Program to understand the use of fprintf ( )**

```
#include<stdio.h>
struct student
{
    char name[10];
    float marks;
}stu;
int main( )
{
    FILE *fp;
    int i,n;
    fp=fopen("students.dat","w") ;
    printf("Enter number of records: ") ;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("Enter name and marks: " ) ;
        scanf("%s%f",stu.name,&stu.marks);
        fprintf (fp,"%s %f\n", stu.name, stu.marks);
    }
    fclose(fp) ;
    return 0;
}
```

**Program 10 Program to understand the use of fscanf( )**

```
#include<stdio.h>
struct student
{
    char name[10];
    float marks;
}stu;
int main( )
{
    FILE *fp;
    int i,n;
    fp=fopen("students.dat","r") ;
    printf ("NAME\tMARKS\n");
    while(fscanf(fp, "%s%f",stu.name,&stu.marks) !=EOF)
        printf("%s\t%f\n",stu.name,stu.marks) ;
    fclose(fp) ;
    return 0;
}
```

**Program 11 WAP to understand the use of fwrite( )**

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

```

struct record{
    char name[20] ;
    int roll;
    float marks;
}student;
int main( )
{
    FILE *fp;
    int i,n;
    fp=fopen("student.dat","wb") ;
    if(fp==NULL){
        printf("Error in opening file..");
        exit(1);
    }
    printf("Enter number of records: ");
    scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("Enter name: ") ;
        scanf("%s",student.name) ;
        printf("Enter roll no: ") ;
        scanf("%d",&student.roll);
        printf("Enter marks: ") ;
        scanf("%f",&student.marks);
        fwrite(&student,sizeof(student),1,fp);
    }
    fclose(fp) ;
    return 0;
}

```

**Program 12 WAP to understand the use of fread( )**

```

#include<stdio.h>
struct record{
    char name[20] ;
    int roll;
    float marks;
}student;
int main( )
{
    FILE *fp;
    int i,n;
    fp=fopen("student.dat","rb") ;
    if(fp==NULL){
        printf("Error in opening file..");
        exit(1);
    }
}

```

```

}
printf("\nNAME\tROLLNO\tMARKS\n");
while(fread(&student,sizeof(student),1,fp)==1)
{
    printf("%s\t",student.name) ;
    printf("%d\t",student.roll) ;
    printf("%f\n",student.marks) ;
}
fclose(fp) ;
return 0;
}

```

**Program 13 WAP to count the number of words in a text file. Assume that a word ends with a space, tab, newline, comma, fullstop, semicolon, colon, hyphen.**

**Program 14 Write a program to copy a file to another file such that blank lines are not written to the new file.**

**Program 15 Write a program to convert all the lower case characters of a file to upper case.**

**Program 16 Write a program to display the total number of alphabets and numeric characters in a file.**

**Program 17 Write a program to remove all comments from the file. Assume that a comment starts with double slash ( //) and continues till the end of the line**

