

Exercise 1:

1. Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.

Program:

```
#include<stdio.h>
void main()
{
    printf("#####\n");
    printf("#\n");
    printf("#\n");
    printf("####\n");
    printf("#\n");
    printf("#\n");
}
```

Output:

```
#####
#
#
####
#
#
```

2. Write a C program to compute the perimeter and area of a rectangle with a height of 7 inches and width of 5 inches.

Program:

```
#include<stdio.h>
void main()
{
    /*Given length and width of rectangle are 7 inches and 5 inches */
    int l=7,b=5,Area;
    Area=l*b;
    printf("Area of the Rectangle is : %d sq inches\n",Area);
}
```

Output:

```
Area of the Rectangle is : 35 sq inches
```

3. Write a C program to display multiple variables.

Program:

```
#include<stdio.h>
void main()
{
    /*Consider initialized variables of different types*/
    /* The intention of this program is to introduce different format specifiers*/
    int n=5;
    float f=8.3761;
    char ch='R';
    double db=6.53421567432;
```

```

printf("Values of variables are as follows:\n");
printf("Integer Variable n holds:%d\n",n);
printf("Floating Point Variable f holds:%f\n",f);
printf("Character Variable ch holds:%c\n",ch);
printf("Double precision Variable db holds:%lf\n",db);
}

```

Output:

Values of variables are as follows:

Integer Variable n holds:5

Floating Point Variable f holds:8.376100

Character Variable ch holds:R

Double precision Variable db holds:6.534216 (Output Restricted to 6 digits after decimal point)

Exercise 2:

1. Write a C program to calculate the distance between the two points.

Program:

```

/* This program computes the distance between two points in xy-plane*/
#include<stdio.h>
#include<math.h>
void main()
{
    float x1,x2,y1,y2;
    float dist;
    printf("Enter x and y co-ordinates of first point:\n");
    scanf("%f%f",&x1,&y1);
    printf("Enter x and y co-ordinates of second point:\n");
    scanf("%f%f",&x2,&y2);
    dist=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    printf("Distance between the given points is: %f",dist);
}

```

Output:

Enter x and y co-ordinates of first point:

2.1 3.2

Enter x and y co-ordinates of second point:

5.34 4.69

Distance between the given points is: 3.566189

2. Write a C program that accepts 4 integers' p, q, r and s from the user, where r and s are positive and p is even. If q is greater than r and s is greater than p and if the sum of r and s is greater than the sum of p and q print "Correct values", otherwise print "Wrong values".

Program:

```

#include<stdio.h>
void main()
{
    int p,q,r,s;
    printf(" Enter positive integers for variables r and s:\n");
}

```

```

scanf("%d%d",&r,&s);
printf("Enter any positive even integer for p:\n");
scanf("%d",&p);
printf("Enter any integer for q:\n");
scanf("%d",&q);
if(q>r && s>p && r+s > p+q)    printf("Correct Values\n");
else printf("Wrong Values\n");
}

```

Output:

Run1:

Enter positive integers for variables r and s:

5 9

Enter any positive even integer for p:

10

Enter any integer for q:

5

Wrong Values

Run2:

Enter positive integers for variables r and s:

5 10

Enter any positive even integer for p:

6

Enter any integer for q:

7

Correct Values

Exercise 3:

1. Write a C program to convert a string to a long integer.

In the C Programming Language, the **strtol** function converts a string to a long integer.

The strtol function skips all white-space characters at the beginning of the string, converts the subsequent characters as part of the number, and then stops when it encounters the first character that isn't a number.

Syntax

The syntax for the strtol function in the C Language is:

```
long int strtol(const char *nptr, char **endptr, int base);
```

Parameters or Arguments

nptr A pointer to a string to convert to a long integer.

endptr It is used by the strtol function to indicate where the conversion stopped. The strtol function will modify *endptr* (if *endptr* is not a null pointer) so that *endptr* points to the first character that was not converted.

base The base of the number being converted. If *base* is between 2 and 36, it is used as the radix of the number. If *base* is zero, the number is assumed to be decimal unless the converted number starts with O (for Octal), Ox (for hex) or OX (for hex).

Program:

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

```

#include<stdlib.h>
int main ()
{
    char buffer[] = "2016 40a0b0 -1101110100110111100110 0x5abfff";
    char * ptr_end;
    long int i1, i2, i3, i4;

    i1 = strtol (buffer,&ptr_end,10);
    i2 = strtol (ptr_end,&ptr_end,16);
    i3 = strtol (ptr_end,&ptr_end,2);
    i4 = strtol (ptr_end,NULL,0);
    printf ("\nIn decimals: %ld, %ld, %ld, %ld.\n\n", i1, i2, i3, i4);
    return 0;
}

```

Output:

In decimals: 2016, 4235440, -3624422, 5947391.

2. Write a Menu-Driven program in C to compute the area of the various geometrical shapes.

Program:

```

#include<stdio.h>
#include<math.h>
void main()
{
    float a,b,c;
    int choice;
    char ch;
    printf(" Menu:\n 1.Rectangle\n 2.Triangle\n 3.Trapezium\n 4.Circle\n\n Enter your Choice\n");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            printf("Enter Length and Breadth of Rectangle:\n");
            scanf("%f%f",&a,&b);
            printf("Area=%f\n",a*b);
            printf("Do you want to continue? Y or N ?\n");
            fflush(stdin);
            scanf("%c",&ch);
            if(ch=='N') break;
        case 2:
            printf("Enter Side of Equilateral Triangle:\n");
            scanf("%f",&a);
            printf("Area=%f\n",sqrt(3)*a*a/4);
            printf("Do you want to continue? Y or N ?\n");
            fflush(stdin);
            scanf("%c",&ch);
            if(ch=='N') break;
        case 3:
            printf("Enter lengths of parallel sides and distance between parallel sides of Trapezium\n");
            scanf("%f%f%f",&a,&b,&c);
            printf("Area=%f\n",c*(a+b)/2);
            printf("Do you want to continue? Y or N ?\n");
            fflush(stdin);
            scanf("%c",&ch);
            if(ch=='N') break;
        case 4:
            printf("Enter the radius of Circle\n");
            scanf("%f",&a);
            printf("Area=%f\n",22*a*a/7);
    }
}

```

Output:

Menu:

- 1.Rectangle
- 2.Triangle
- 3.Trapezium
- 4.Circle

Enter your Choice

2

Enter Side of Equilateral Triangle:

6

Area=15.588457

Do you want to continue? Y or N ?

Y

Enter lengths of parallel sides and distance between parallel sides of Trapezium

7 11 5

Area=45.000000

Do you want to continue? Y or N ?

N

3. Write a C program to calculate the factorial of a given number.

Program:

```
#include<stdio.h>
void main()
{
    int n,i,f=1;
    printf("Enter any positive ineger:\n");
    scanf("%d",&n);
    if(n==0)f=1;
    else
        for(i=2;i<=n;i++) f=f*i;
    printf("Factorial of %d is %d\n ",n,f);
}
```

Output:

Enter any positive ineger:

6

Factorial of 6 is 720

Exercise 4:

1. Write a program in C to display the n terms of even natural number and their sum.

Program:

```
#include<stdio.h>
void main()
{
    int n,i=1,j=2,sum=0;
    printf("How many even natural numbers to print?\n");
    scanf("%d",&n);
    printf("First %d even natural numbers are:\n",n);
    for(;i<=n;)
    {
        printf(" %d ",j);
        sum=sum+j;
        j=j+2;
    }
}
```

```

        i++;
    }
    printf("\nSum of the above terms is %d\n",sum);
}

```

Output:

How many even natural numbers to print?

6

First 6 even natural numbers are:

2 4 6 8 10 12

Sum of the above terms is 42

2. Write a program in C to display n terms of the harmonic series and their sum. $1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$ terms.

Program:

```

#include<stdio.h>
void main()
{
    int n,i=2;
    float sum=1.0;
    printf("How many terms of Harmonic series are required?\n");
    scanf("%d",&n);
    printf("1+");
    for(;i<=n;i++)
    {
        printf("1/%d",i);
        if(i!=n)printf("+");
        sum=sum+1.0/i;
    }
    printf("\nSum of above series is: %f\n",sum);
}

```

Output:

How many terms of Harmonic series are required?

5

$1+1/2+1/3+1/4+1/5$

Sum of above series is: 2.283334

3. Write a C program to check whether a given number is an Armstrong number or not.

Program:

```

#include<stdio.h>
#include<math.h>
void main()
{
    int n,m,sum=0,digit;
    printf("Enter any positive integer:\n");
    scanf("%d",&n);
    m=n;

```

```

while(n!=0)
{
    digit=n%10;
    sum+=pow(digit,3);
    n/=10;
}
if(sum==m) printf("Given integer is Armstrong\n");
else printf("Given integer is Not Armstrong\n");
}

```

Output:

Run1:

Enter any positive integer:

155

Given integer is Not Armstrong

Run2:

Enter any positive integer:

371

Given integer is Armstrong

Exercise 5:

1. Write a program in C to print all unique elements in an array.

Program:

```

#include<stdio.h>
void main()
{
    int n,i,a[10],f,j;
    printf("Enter size of Array:\n");
    scanf("%d",&n);
    printf("Enter %d integer elements\n",n);
    for(i=0;i<n;i++) scanf("%d",&a[i]);
    printf("Unique elements of the array are:\n");
    /*Following loop prints unique elements considering last position where
    a repeated element is found*/
    for(i=0;i<n;i++)
    {
        f=0;
        for(j=i+1;j<n;j++)
            if(a[i]==a[j])f=1;
        if(f==0)printf("%d[%d] ",a[i],i+1);
    }
}

```

Output:

Enter size of Array:

9

Enter 9 integer elements

1 1 3 2 1 5 4 3 5

Unique elements of the array are:

2[4] 1[5] 4[7] 3[8] 5[9] (Subscripts indicate the most recent position where a repeated element is found)

2. Write a program in C to separate odd and even integers in separate arrays.

Program:

```
#include<stdio.h>
void main()
{
    int n,i,j=0,k=0,a[10],even[10],odd[10];
    printf("Enter size of Array:\n");
    scanf("%d",&n);
    printf("Enter %d integer elements\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        if(a[i]%2==0) even[j++]=a[i];
        else odd[k++]=a[i];
    }
    if(j==0)printf("No Even elements\n");
    else
    {
        printf("Even Array is:\n");
        for(i=0;i<j;i++) printf("%d ",even[i]);
    }
    if(k==0)printf("\nNo Odd elements\n");
    else
    {
        printf("\nOdd Array is:\n");
        for(i=0;i<k;i++) printf("%d ",odd[i]);
    }
}
```

Output:

Enter size of Array:

8

Enter 8 integer elements

-2 -3 4 3 1 6 8 7

Even Array is:

-2 4 6 8

Odd Array is:

-3 3 1 7

3. Write a program in C to sort elements of array in ascending order.

Program:

```
#include<stdio.h>
void main()
{
    int n,i,j,a[10],temp;
    printf("Enter size of array:\n");
    scanf("%d",&n);
    printf("Enter %d integer elements:\n",n);
    for(i=0;i<n;i++) scanf("%d",&a[i]);
    printf("Array Sorted in ascending order is:\n");
    /*Selection Sort*/
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
    for (i=0;i<n;i++) printf("%d ",a[i]);
}
```

Output:

Enter size of array:

8

Enter 8 integer elements:

-2 0 3 -1 4 5 4 6

Array Sorted in ascending order is:

-2 -1 0 3 4 4 5 6

Exercise 6:

1. Write a program in C for multiplication of two square Matrices.

Program:

```
#include<stdio.h>
void main()
{
    int a[4][4],b[4][4],c[4][4],i,j,k,n;
    printf("Enter size of square matrix\n");
    scanf("%d",&n);
    printf("Enter %dX%d elements of matrix A:\n",n,n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++) scanf("%d",&a[i][j]);
    printf("Enter %dX%d elements of matrix B:\n",n,n);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++) scanf("%d",&b[i][j]);
    for(i=0;i<n;i++)
```

```

        for(j=0;j<n;j++)
        {
            c[i][j]=0;
            for(k=0;k<n;k++)c[i][j]+=a[i][k]*b[k][j];
        }
        printf("Multiplication of Matrices A and B is:\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            printf("%d ",c[i][j]);
            printf("\n");
        }
    }
}

```

Output:

```

Enter size of square matrix
2
Enter 2X2 elements of matrix A:
-1 1
1 -1
Enter 2X2 elements of matrix B:
1 2
-2 1
Multiplication of Matrices A and B is:
-3 -1
3 1

```

2. Write a program in C to find transpose of a given matrix.

Program:

```

#include<stdio.h>
void main()
{
    int m,n,i,j,a[5][5];
    printf("Enter size of matrix maximum 5X5\n");
    scanf("%d%d",&m,&n);
    printf("Enter %dX%d elements:\n",m,n);
    for(i=0;i<m;i++)
        for(j=0;j<n;j++) scanf("%d",&a[i][j]);
    printf("transpose of given matrix is:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++) printf("%d ",a[j][i]);
        printf("\n");
    }
}

```

Output:

```

Enter size of matrix maximum 5X5
4 3

```

Enter 4X3 elements:

1 2 3

4 3 1

0 -3 -4

-1 3 8

Transpose of given matrix is:

1 4 0 -1

2 3 -3 3

3 1 -4 8

Exercise 7:

1. Write a program in C to search an element in a row wise and column wise sorted matrix.

Program:

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

```
void main()
```

```
{
    int i,j,k,m,n,a[5][5],se,f=0;
    printf("Enter size of matrix maximum 5x5:\n");
    scanf("%d%d",&m,&n);
    printf("Enter %dX%d elements:\n",m,n);
    printf("Enter elements in row wise and column wise sorted order:\n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++) scanf("%d",&a[i][j]);
    printf("Enter Search Element:\n");
    scanf("%d",&se);
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(a[i][j]==se)
            {
                f=1;
                break;
            }
            if(a[i][j]>se)
            {
                k=1;
                break;
            }
        }
        if(f==1 || k==1) break;
    }
    if(f==0) printf("Element not found\n");
    else printf("Element Found\n");
}
```

Output:

Run1:

Enter size of matrix maximum 5x5:

2 3

Enter 2X3 elements:

Enter elements in row wise and column wise sorted order:

1 3 5

3 6 8

Enter Search Element:

6

Element Found

Run2:

Enter size of matrix maximum 5x5:

2 3

Enter 2X3 elements:

Enter elements in row wise and column wise sorted order:

1 3 5

3 6 8

Enter Search Element:

7

Element not found

2. Write a program in C to print individual characters of string in reverse order.

Program:

```
#include<stdio.h>
void main()
{
    char s[10];
    int l=0,i;
    printf("Enter any Stringof less than 10 characters\n");
    gets(s);
    while(s[l++]!='\0');
    printf("Individual Characters in the reverse order are:\n");
    for(i=l-2;i>=0;i--) printf("%c ",s[i]);
}
```

Output:

Enter any Stringof less than 10 characters

Rose Milk

Individual Characters in the reverse order are:

k l i M e s o R

Exercise 8:

1. Write a program in C to compare two strings without using string library functions.

Program:

```
#include<stdio.h>
void main()
{
    char s1[10],s2[10];
    int l1=0,diff=0;
    printf("Enter first string less than 10 characters:\n");
    gets(s1);
    printf("Enter second string less than 10 characters:\n");
    gets(s2);
```

```

while(s1[l1]!='\0' || s2[l1]!='\0')
{
    if(s1[l1]=='\0' && s2[l1]!='\0')
    {
        diff=-1;
        break;
    }
    if(s1[l1]!='\0' && s2[l1]=='\0')
    {
        diff=1;
        break;
    }
    if(s1[l1]!=s2[l1])
    {
        diff=s1[l1]-s2[l1];
        printf("%d\n",diff);
        break;
    }
    else l1++;
}
if(diff==0) printf("Strings are equal\n");
else if(diff>0) printf("String1 is greater\n");
else printf("String2 is greater\n");
}

```

Output:

Run1:

Enter first string less than 10 characters:

Ros

Enter second string less than 10 characters:

Ros

Strings are equal

Run2:

Enter first string less than 10 characters:

Sun

Enter second string less than 10 characters:

rose

-31

String2 is greater

2. Write a program in C to copy one string to another string.

Program:

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

```
void main()
```

```

{
    char s1[10],s2[10];
    int l=0,i=0;
    printf("Enter any string without spaces\n");
    scanf("%s",s1);
    while(s1[l++]!='\0') s2[i++]=s1[l-1];
    s2[i]='\0';
}

```

```

        printf("The copied string is %s",s2);
    }

```

Output:

Enter any string without spaces

TigeR

The copied string is TigeR

Exercise 9:

1. Write a C Program to Store Information Using Structures with Dynamically Memory Allocation

Program:

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    struct student
    {
        int s_no;
        char sname[10];
        float gpa;
    };
    int n,i;
    struct student *ptr;
    printf("How many structure variables needed?\n");
    scanf("%d",&n);
    ptr= (struct student*)malloc(n*sizeof(struct student));
    printf("Enter %d student details:\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d%s%f",&(ptr+i)->s_no,(ptr+i)->sname,&(ptr+i)->gpa);
    }
    printf("The details of students are:\n");
    for(i=0;i<n;i++)
    {
        printf("%d %s %f\n",(ptr+i)->s_no,(ptr+i)->sname,(ptr+i)->gpa);
    }
    free(ptr);
}

```

Output:

How many structure variables needed?

2

Enter 2 student details:

101 Ramesh 7.86

102 Smith 8.21

The details of students are:

101 Ramesh 7.860000

102 Smith 8.210000

2. Write a program in C to demonstrate how to handle the pointers in the program.

Program:

```
/*Program to illustrate assignment and accessing data through pointers*/
#include<stdio.h>
void main()
{
    int n1=5,n2,*iptr;
    float f1,f2,*fptr;
    void *vptr;
    vptr=&n1;/* Void Pointer can receive address of any type of variable*/
    printf("Value of n1=%d %d\n",n1,*(int*)vptr);
    iptr=&n2;
    *iptr=7+*(int*)vptr;
    printf("Value of n2=%d\n",*iptr);
    f1=9.187534;
    fptr=&f1;
    printf("Value of f1=%f\n",*fptr);
    printf("Address of fptr=%p\n",&fptr);
    printf("Address in fptr=%p\n",fptr);
    printf("Address obtained by pointer increment of fptr=%p\n",++fptr);
}
```

Output:

```
Value of n1=5 5
Value of n2=12
Value of f1=9.187534
Address of fptr=000000000062FDF8
Address in fptr=000000000062FE04
Address obtained by pointer increment of fptr=000000000062FE08
```

Exercise 10:

*1. Write a program in C to demonstrate the use of & (address of) and *(value at address) operator.*

Program:

```
#include<stdio.h>
void main()
{
    int n,*ptr;
    printf("Enter any integer value:\n");
    scanf("%d",&n);/*Reading value from key board at the address of a variable*/
    ptr=&n;/*Assigning the address of a variable to a pointer variable*/
    printf("Value of n is:%d\n",n);
    printf("value of n is:%d\n",*ptr);/* Dereferencing using * operator*/
}
```

Output:

```
Enter any integer value:
6
Value of n is:6
value of n is:6
```

2. Write a program in C to add two numbers using pointers.

Program:

```
#include<stdio.h>
void main()
{
    int m,n,sum;
    int *p1,*p2,*p3;
    printf("Enter any two integers:\n");
    scanf("%d%d",&m,&n);
    p1=&m;
    p2=&n;
    p3=&sum;
    *p3=*p1+*p2;
    printf("Sum of given numbers is:%d\n",*p3);
}
```

Output:

```
Enter any two integers:
6 14
Sum of given numbers is:20
```

Exercise 11:

1. Write a program in C to add numbers using call by reference.

Program:

```
#include<stdio.h>
int add(int*,int*);
void main()
{
    int m,n,sum;
    printf("Enter any two integers:\n");
    scanf("%d%d",&m,&n);
    sum=add(&m,&n);
    printf("Sum of givn numbers is:%d\n",sum);
}
int add(int *x,int *y)
{
    return (*x+*y);
}
```

Output:

```
Enter any two integers:
7 18
Sum of givn numbers is:25
```

2. Write a program in C to find the largest element using Dynamic Memory Allocation.

Program:

```
#include<stdio.h>
#include<stdlib.h>
int large(int*,int);
void main()
{
```



```

        int n,i,big,*ptr;
        printf("How many integer elements to store?\n");
        scanf("%d",&n);
        ptr=(int*)malloc(n*sizeof(int));
        printf("Enter %d elements:\n",n);
        for(i=0;i<n;i++)scanf("%d",ptr+i);
        big=large(ptr,n);
        printf("The largest element among given elements is:%d\n",big);
        free(ptr);
    }
    int large(int *m,int l)
    {
        int i,big;
        big=*m;
        for(i=1;i<l;i++)
            if(*(m+i)>big) big=*(m+i);
        return big;
    }

```

Output:

```

How many integer elements to store?
6
Enter 6 elements:
3 -1 2 6 4 5
The largest element among given elements is:6

```

Exercise 12:

1. Write a program in C to swap elements using call by reference.

Program:

```

#include<stdio.h>
void swap(int*,int*);
void main()
{
    int m,n;
    printf("Enter any two integers:\n");
    scanf("%d%d",&m,&n);
    printf("The elements before swapping are: %d and %d\n",m,n);
    swap(&m,&n);
    printf("The elements after swapping are: %d and %d\n",m,n);
}
void swap(int *p,int *q)
{
    int t;
    t=*p;
    *p=*q;
    *q=t;
}

```

Output:

```

Enter any two integers:

```

7 10

The elements before swapping are: 7 and 10

The elements after swapping are: 10 and 7

2. Write a program in C to count the number of vowels and consonants in a string using a pointer.

Program:

```
#include<stdio.h>
void main()
{
    char str[10],*ptr;
    int c_vowel=0,c_conso=0;
    printf("Enter any string less than 10 characters\n");
    gets(str);
    ptr=str;
    while(*ptr!='\0')
    {
        if(isalpha(*ptr))
        {
            *ptr=toupper(*ptr);
            if(*ptr=='A' || *ptr=='E' || *ptr=='I' || *ptr=='O' || *ptr=='U')c_vowel++;
            else c_conso++;
        }
        ptr++;
    }
    printf("Count of Vowels=%d\n",c_vowel);
    printf("Count of Consonants=%d\n",c_conso);
}
```

Output:

Enter any string less than 10 characters

Ravindra1*

Count of Vowels=3

Count of Consonants=5

Exercise 13:

1. Write a program in C to show how a function returning pointer.

Program:

```
#include <stdio.h>
int* area(int,int);
int main()
{
    int *p,x,y;
    printf("Enter length and breadth of rectangle:\n");
    scanf("%d%d",&x,&y);
    p=area(x,y);
    printf("Area=%d\n",*p);
}
```

```
int* area(int m,int n)
{
    static int a;
    a=m*n;
    return (&a);
}
```

Output:

Enter length and breadth of rectangle:

7 9

Area=63

2. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using malloc() function.

Program:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,sum=0,*ptr;
    printf("Enter number of elements to add:\n");
    scanf("%d",&n);
    ptr=(int*)malloc(n*sizeof(int));
    for(i=0;i<n;i++) printf("%d ",*(ptr+i));
    printf("\nEnter %d Elements\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",ptr+i);
        sum+=*(ptr+i);
    }
    printf("Sum of elements is:%d\n",sum);
    printf("The addresses of the elements are:\n");
    for(i=0;i<n;i++) printf("%p\n",ptr+i);
    free(ptr);
}
```

Output:

Enter number of elements to add:

6

1709088 0 1704272 0 0 0

Enter 6 Elements

6 1 3 2 4 1

Sum of elements is:17

The addresses of the elements are:

00000000001A6BD0

00000000001A6BD4

00000000001A6BD8

00000000001A6BDC

00000000001A6BE0
00000000001A6BE4

Exercise 14:

1. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using calloc() function. Understand the difference between the above two programs.

Program:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,sum=0,*ptr;
    printf("Enter number of elements to add:\n");
    scanf("%d",&n);
    ptr=(int*)calloc(n,sizeof(int));
    for(i=0;i<n;i++) printf("%d ",*(ptr+i));
    printf("\nEnter %d Elements\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",ptr+i);
        sum+=*(ptr+i);
    }
    printf("Sum of elements is:%d\n",sum);
    printf("The addresses of the elements are:\n");
    for(i=0;i<n;i++) printf("%p\n",ptr+i);
    free(ptr);
}
```

Output:

```
Enter number of elements to add:
6
0 0 0 0 0 0
Enter 6 Elements
5 6 2 3 4 1
Sum of elements is:21
The addresses of the elements are:
0000000000A36BD0
0000000000A36BD4
0000000000A36BD8
0000000000A36BDC
0000000000A36BE0
0000000000A36BE4
```

The differences observed between the use of malloc() and calloc() functions are:

- i) malloc() initializes elements to garbage values. calloc() initializes all elements to zero.
- ii) malloc() accepts one parameter. calloc() accepts two parameters.

2. Write a program in C to convert decimal number to binary number using the function.

Program:

```
#include<stdio.h>
void convert(int, int);
int main()
{
    int num,base;
    printf("Enter a positive decimal number : ");
    scanf("%d", &num);
    printf("Enter base of the system you want:\n");
    scanf("%d",&base);
    switch(base)
    {
        case 2:
            printf("\nBinary number :: ");
            convert(num, 2);
            printf("\n");
            break;
        case 8:
            printf("\nOctal number :: ");
            convert(num, 8);
            printf("\n");
            break;
        case 16:
            printf("\nHexadecimal number :: ");
            convert(num, 16);
            printf("\n");
            break;
        default:
            printf("Invalid choice\n");
    }
}
void convert (int num, int base)
{
    int rem = num%base;
    if(num==0)return;
    convert(num/base, base);
    if(rem < 10)printf("%d", rem);
    else printf("%c", rem-10+'A' );
}
```

Output:

Enter a positive decimal number : 654

Enter base of the system you want:

2

Binary number :: 1010001110

Exercise 15:

1. Write a program in C to check whether a number is a prime number or not using the function.

Program:

```
#include<stdio.h>
int prime(int);
void main()
{
    int n,f;
    printf("Enter any positive integer:\n");
    scanf("%d",&n);
    f=prime(n);
    if(f==0)printf("\nPrime");
    else printf("Not a Prime\n");
}
int prime(int m)
{
    int c=0,i=2;
    while(i<=m/2)
    {
        if(m%i==0)
        {
            c=1;
            break;
        }
        else i++;
    }
    return c;
}
```

Output:

Run1:

Enter any positive integer:

53

Prime

Run2:

Enter any positive integer:

57

Not a Prime

2. Write a program in C to get the largest element of an array using the function.

Program:

```
#include<stdio.h>
int large(int[],int);
void main()
{
    int n,i,a[10],big;
    printf("Size of the array?\n");
    scanf("%d",&n);
    printf("Enter %d elements:\n",n);
    for(i=0;i<n;i++)scanf("%d",&a[i]);
    big=large(a,n);
    printf("The largest element among given elements is:%d\n",big);
}
```

```

int large(int m[10],int l)
{
    int i,big;
    big=m[0];
    for(i=1;i<l;i++)
        if(m[i]>big) big=m[i];
    return big;
}

```

Output:

Size of the array?

8

Enter 8 elements:

4 2 1 5 2 7 5 1

The largest element among given elements is:7

Exercise 16:

1. Write a program in C to append multiple lines at the end of a text file.

Program:

```

#include <stdio.h>
int main()
{
    FILE *fp;
    int i,n;
    char str[50];
    char fname[20];
    char str1;

    printf("Enter the file name to which to append the lines\n");
    scanf("%s",fname);
    fp= fopen(fname,"a");
    printf("How many lines to append?\n");
    scanf("%d",&n);
    printf("Enter the lines: \n");
    for(i=0;i<=n;i++)
    {
        fgets(str, sizeof(str),stdin);
        fputs(str, fp);
        fflush(stdin);
    }
    fclose(fp);
    /*Display appended file*/
    fp=fopen (fname,"r");
    printf("\n The contents of the file %s are :\n",fname);
    str1 = fgetc(fp);
    while (str1 != EOF)
    {
        printf ("%c", str1);
        str1 = fgetc(fp);
    }
    printf("\nEnd of file");
}

```

```
fclose (fp);  
}
```

Output:

Enter the file name to which to append the lines

Student

How many lines to append?

3

Enter the lines:

Practice makes a person perfect.

Good Programming is a skill.

C is procedural language.

The contents of the file Student are :

Sujith is the topper with 8.67

Practice makes a person perfect.

Good Programming is a skill.

C is procedural language.

End of file

2. Write a program in C to copy a file in another name.

Program:

```
#include<stdio.h>  
#include<stdlib.h>  
void main()  
{  
    FILE *fp1,*fp2;  
    char fname1[10],fname2[10],ch;  
    printf("Enter source file name\n");  
    scanf("%s",fname1);  
    fp1=fopen(fname1,"r");  
    if(fp1==NULL)  
    {  
        printf("File does not exist or error in opening the file\n");  
        exit(1);  
    }  
    printf("Enter the new file name:\n") ;  
    scanf("%s",fname2);  
    fp2=fopen(fname2,"w");  
    printf("The contents of original file are:\n");  
    while(1)  
    {  
        ch=fgetc(fp1);  
        printf("%c",ch);  
        fputc(ch,fp2);  
        if(ch==EOF) break;  
    }  
    fclose(fp1);  
    fclose(fp2);  
}
```



```

        fp1=fopen(fname2,"r");
        printf("\nThe new file after copy is:\n");
        while((ch=fgetc(fp1))!=EOF) printf("%c",ch);
        printf("\nEnd of file.");
        fclose(fp1);
    }

```

Output:

Enter source file name

Student

Enter the new file name:

Std_New

The contents of original file are:

Sujith is the topper with 8.67

Practice makes a person perfect.

Good Programming is a skill.

C is procedural language.

The new file after copy is:

Sujith is the topper with 8.67

Practice makes a person perfect.

Good Programming is a skill.

C is procedural language.

End of file.

3. Write a program in C to remove a file from the disk.

Program:

```

#include <stdio.h>
void main()
{
    int status;
    char fname[20];
    printf(" Input the name of the file to delete:\n");
    scanf("%s",fname);
    status=remove(fname);
    if(status==0)
        printf(" The file %s is deleted successfully..!!\n",fname);
    else
        printf(" Unable to delete file %s\n\n",fname);
}

```

Output:

Run1:

Input the name of the file to delete:

Std_New

Unable to delete file Std_New

Run2:

Input the name of the file to delete:

New_Std

The file New_Std is deleted successfully...!!