**01\_Arithmetic\_Operations.c**

**02\_Multiplication\_Table.c**

**03\_reverse.c**

**04\_pattern1.c**

**05\_Pattern2.c**

**06\_pattern3.c**

**07\_pattern4.c**

**08\_pattern5.c**

**09\_format.c**

**10\_distance.c**

**11\_relational.c**

**12\_areas.c**

**13\_factorial.c**

**14\_evennumbers.c**

**15\_harmonic.c**

**16\_armstrong.c**

**17\_unique.c**

**18\_ multi\_sq\_Matrices.c**

**19\_ transpose\_Matrices.c**

**20\_string\_reverse.c**

**21\_string\_copy.c**

**22\_prime\_function.c**

**23\_swap\_refere.c**

**24\_sum\_pointer.c**

**25\_palindrome.c**

**26\_gcd.c**

**27\_odd\_even\_sep\_array.c**

**28\_ ascending.c**

**29\_matrix\_sorting.c**

**30\_ string\_rever.c**

**31\_string\_comparison.c**

**32\_structures.c**

**33\_pointers.c**

**34\_pointer\_addr\_value.c**

**35\_sum\_refer.c**

**36\_max\_dyn\_mem.c**

**37\_vowels\_conso.c**

**38\_func\_pointer.c**

**39\_sum\_malloc.c**

**40\_sum\_callocmem.c**

**41\_deci\_bina.c**

**42\_deci\_hexa.c**

**43\_ max\_func.c**

**44\_append.c**

**45\_file\_copy.c**

**46\_file\_remove.c**

**47\_series.c**

**48\_numbers.c**

**49\_standard\_deviation.c**

**50\_array1d.c**

**51\_matrices.c**

**52\_string.c**

**53\_fact\_fib\_gcd.c**

**53\_numbers\_.c**

**54\_search.c**

**55\_data.c**

**56\_complex**

**57\_calander.c**

**58\_prime.c**

**59\_binary\_fun.c**

**60\_deci\_bina\_func**

**61\_deci\_octal.c**

**62\_octa\_deci.c**

**63\_first.c**

**64\_hcf.c**

**65\_hcf\_gcd.c**

**66\_hexa\_deci.c**

**67\_lcm.c**

**68\_lcm\_.c**

**hello.c**

**1)/\*\*\* Arithmetic Operations \*\*\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,c;

clrscr();

printf("\n Enter any two Numbers:");

scanf("%d%d",&a,&b);

printf("\n%d + %d = %d",a,b,a+b);

printf("\n%d - %d = %d",a,b,a-b);

printf("\n%d \* %d = %d",a,b,a\*b);

printf("\n%d / %d = %d",a,b,a/b);

printf("\n%d mod %d = %d",a,b,a%b);

getch();

}

**Output:**

Enter any two Numbers:5

6

5 + 6 = 11

5 - 6 = -1

5 \* 6 = 30

5 / 6 = 0

5 mod 6 = 5

**2)/\*\*\* Multiplication Table \*\*\*/**

#include<stdio.h>

void main()

{

int i,n;

clrscr();

printf("\n Enter the Multiplication Table to Print:");

scanf("%d",&n);

for(i=1;i<=10;i++)

printf("\n %d x %d = %d",n,i,n\*i);

getch();

}

**Output:**

Enter the Multiplication Table to Print:2

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

2 x 6 = 12

2 x 7 = 14

2 x 8 = 16

2 x 9 = 18

2 x 10 = 20

**3)/\*\*\* Reverse of a given Number \*\*\*\*/**

#include<stdio.h>

void main()

{

int n,remainder,rev;

clrscr();

printf("\n Enter any number:");

scanf("%d",&n);

rev=0;

while(n!=0)

{

remainder=n%10;

n=n/10;

rev=rev\*10+remainder;

}

printf("\n Reverse of Given Number = %d",rev);

getch();

}

**Output:**

Enter any number:234

Reverse of Given Number = 432

**4) /\* Print Required Pattern \*/**

#include<stdio.h>

void main()

{ inti,j,n;

clrscr();

printf("\n Enter Number of Rows :");

scanf("%d",&n);

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

{

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

printf(" \*");

printf("\n"); }

getch();}

**Output**:

Enter Number of Rows :5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**5) /\* Print Required Pattern \*/**

#include<stdio.h>

void main()

{int i,j,n;

clrscr();

printf("\n Enter Number of Rows :");

scanf("%d",&n);

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

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

printf("%3d",j);

printf("\n");}

getch();}

**Output:**

Enter Number of Rows :4

1

1 2

1 2 3

1 2 3 4

**6)/\* Print Required Pattern \*/**

#include<stdio.h>

void main()

{

int i,j,n;

clrscr();

printf("\n Enter Number of Rows :");

scanf("%d",&n);

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

{

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

printf("%3d",i);

printf("\n");

}

getch();

}

**OutPut:**

Enter Number of Rows :7

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

6 6 6 6 6 6

7 7 7 7 7 7 7

**7.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**

#include <stdio.h>

int main()

{

printf("######\n");

printf("#\n");

printf("#\n");

printf("#####\n");

printf("#\n");

printf("#\n");

printf("#\n");

return(0);

}

**OutPut:**

######

#

#

#####

#

#

#

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

#include<stdio.h>

int main()

{

int height=7,width=5,area,perimeter;

scanf("%d %d ", &height,&width);

area = height\*width;

printf("Area of the rectangle = %d square inches\n", area);

perimeter = 2\*(height+width);

printf("Perimeter of the rectangle = %d inches\n", perimeter);

return 0;

}

**OutPut:**

5

10

Area of the rectangle = 50 square inches

Perimeter of the rectangle = 30 inches

**9.Write a C program to display multiple variables.**

**Program:**

#include<stdio.h>

int main()

{

char a='d';

int b=2021;

float c=2050.23;

double d=12345678;

printf("\n character variable:%c",a);

printf("\n integer variable:%d",b);

printf("\n floating variable:%f",c);

printf("\n double variable:%lf",d);

return 0;

}

**OutPut:**

character variable:d

integer variable:2021

floating variable:2050.229980

double variable:12345678.000000

**10.Calculate the distance between the two points**

#include <stdio.h>

#include <math.h>

int main()

{

float x1,y1,x2,y2,distance;

printf("\nEnter x1 value:");

scanf("%f",&x1);

printf("\nEnter y1 value:");

scanf("%f",&y1);

printf("\nEnter x2 value:");

scanf("%f",&x2);

printf("\nEnter y2 value:");

scanf("%f",&y2);

distance=(x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1);

printf("Distance between two points:%f", sqrt(distance));

printf("\n");

return 0;

}

**OutPut:**

Enter x1 value:2

Enter y1 value:4

Enter x2 value:6

Enter y2 value:8

Distance between two points:5.656854

**11. Write a C program that accepts 4 integers p, q, r, 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".**

#include <stdio.h>

int main()

{int p,q,r,s;

printf("\nEnter the first integer: ");

scanf("%d",&p);

printf("\nEnter the second integer: ");

scanf("%d",&q);

printf("\n 3rd and 4th integers: ");

scanf("%d%d",&r,&s);

**if(q>r && s>p &&r+s>p+q&& r>0 && s>0 && p%2 == 0)**

{

printf("\nCorrect values\n");}

else

{printf("\nWrong values\n");}

return 0;

}

**OutPut:**

Enter the first integer: 10

Enter the second integer: 55

3rd and 4th integers: 7 14

Wrong values

**12.Write a program in C which is a Menu-Driven Program to compute the area of the various geometrical shape.**

#include <stdio.h>

int main ()

{int choice,radius,length;

int breadth,base,height;

float area;

printf("1.Area of circle\n");

printf("2.Area of rectangle\n");

printf("3.Area of triangle\n");

printf("\nEnter your choice:");

scanf("%d",&choice);

switch(choice)

{case 1:

printf("Enter r of the circle:");

scanf("%d",&radius);

area=3.14\*radius\*radius;

break;

case 2:

printf("l and b of the recta:");

scanf("%d%d",&length,&breadth);

area=length\*breadth;

break;

case 3:

printf("b and h of the triangle:");

scanf("%d%d",&base,&height);

area=0.5\*base\*height;

break;

default:

printf("Invalid choice");

break;

}

printf("\nThe area is : %f",area);

return 0;

}

**OutPut:**

1.Area of circle

2.Area of rectangle

3.Area of triangle

Enter your choice:1

Enter r of the circle:2

The area is : 12.560000

1.Area of circle

2.Area of rectangle

3.Area of triangle

Enter your choice:2

l and b of the recta:10 6

The area is : 60.000000

1.Area of circle

2.Area of rectangle

3.Area of triangle

Enter your choice:3

b and h of the triangle:10 4

The area is : 20.000000

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

#include <stdio.h>

int main()

{

int i,fact=1,num;

printf("Enter the number:");

scanf("%d",&num);

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

fact=fact\*i;

printf("The Factorial of %d is: %d\n",num,fact);

}

**OutPut:**

Enter the number:6

The Factorial of 6 is: 720

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

#include <stdio.h>

int main()

{int i,n,sum=0;

printf("Input number of terms : ");

scanf("%d",&n);

printf("\nThe even numbers are :");

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

{

printf("%d ",2\*i);

sum+=2\*i;}

printf("\nThe Sum of even Natural Number upto %d terms : %d \n",n,sum);

return 0;

}

**Output:**

Input number of terms : 5

The even numbers are :2 4 6 8 10

The Sum of even Natural Number upto 5 terms : 30

**15. Write a program in C to display the n terms of harmonic series and their sum. The series is :1 + 1/2 + 1/3 .. 1/n terms**

#include <stdio.h>

int main()

{

int i,n;

float s=0.0;

printf("Input the number of terms : ");

scanf("%d",&n);

printf("\n\n");

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

{if(i<n)

{printf("1/%d + ",i);

s+=1/(float)i;}

if(i==n)

{printf("1/%d ",i);

s+=1/(float)i;} }

printf("\nSum of Series upto %d terms : %f \n",n,s);

return 0;

}

**Output:**

Input the number of terms : 5

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

Sum of Series upto 5 terms : 2.283334

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

#include<stdio.h>

int main()

{

int n,r,sum=0,temp;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{ r=n%10;

sum=sum+(r\*r\*r);

n=n/10; }

if(temp==sum)

printf("armstrong number ");

else

printf("not armstrong number");

return 0;

}

**OutPut:**

enter the number=135

not armstrong number

enter the number=153

armstrong number

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

#include <stdio.h>

int main()

{

int arr1[100], n,ctr, i, j, k;

ctr=0;

printf("\n\nPrint all unique elements of an array:\n");

printf("------------------------------------------\n");

printf("\nEnter the number of elements to be stored in the array: ");

scanf("%d",&n);

printf("Enter %d elements in the array :\n",n);

for(i=0;i<n;i++)

{

printf("element - %d : ",i);

scanf("%d",&arr1[i]);

}

printf("\nThe unique elements found in the array are: \n");

for(i=0; i<n; i++)

{ctr=0;

for(j=0,k=n; j<k+1; j++)

{if (i!=j)

{if(arr1[i]==arr1[j])

{ctr++;}} }

if(ctr==0)

{

printf("%d\t",arr1[i]);

}

}

printf("\n\n");

return 0;

}

**OutPut:**

Print all unique elements of an array:

------------------------------------------

Enter the number of elements to be stored in the array: 10

Enter 10 elements in the array :

element - 0 : 12

element - 1 : 12

element - 2 : 12

element - 3 : 10

element - 4 : 11

element - 5 : 14

element - 6 : 10

element - 7 : 11

element - 8 : 16

element - 9 : 18

The unique elements found in the array are:

14 16 18

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

#include <stdio.h>

int main()

{ int a[50][50],b[50][50],c[50][50],i,j,k,r1,c1,r2,c2,sum=0;

printf("\n\nMultiplication of two Matrices :\n");

printf("----------------------------------\n");

printf("\nEnter the rows and columns of first matrix:");

scanf("%d %d",&r1,&c1);

printf("\nEnter the rows and columns of second matrix:");

scanf("%d %d",&r2,&c2);

if(c1!=r2)

{

printf("Mutiplication of Matrix is not possible.");

printf("\nColumn of first matrix and row of second matrix must be same."); }

else

{

printf("Enter elements in the first matrix:\n");

for(i=0;i<r1;i++)

{for(j=0;j<c1;j++)

{

printf("element - [%d][%d] : ",i,j);

scanf("%d",&a[i][j]);

}

}

printf("Enter elements in the second matrix:\n");

for(i=0;i<r2;i++)

{for(j=0;j<c2;j++)

{printf("element - [%d][%d] : ",i,j);

scanf("%d",&b[i][j]);}}

printf("\nThe First matrix is:\n");

for(i=0;i<r1;i++)

{

printf("\n");

for(j=0;j<c1;j++)

printf("%d\t",a[i][j]);}

printf("\nThe Second matrix is:\n");

for(i=0;i<r2;i++)

{

printf("\n");

for(j=0;j<c2;j++)

printf("%d\t",b[i][j]);}

//multiplication of matrices

for(i=0;i<r1;i++)

for(j=0;j<c2;j++)

c[i][j]=0;

for(i=0;i<r1;i++) //row of first matrix

{for(j=0;j<c2;j++) //column of second matrix

{

sum=0;

for(k=0;k<c1;k++)

sum=sum+a[i][k]\*b[k][j];

c[i][j]=sum;

}

}

printf("\nThe multiplication of two matrices is:\n");

for(i=0;i<r1;i++)

{ printf("\n");

for(j=0;j<c2;j++)

{printf("%d\t",c[i][j]); }}}

printf("\n\n");

return 0;

}

**OutPut:**

Multiplication of two Matrices:

----------------------------------

Enter the rows and columns of first matrix:2 2

Enter the rows and columns of second matrix:2 2

Enter elements in the first matrix:

element - [0][0] : 12

element - [0][1] : 18

element - [1][0] : 65

element - [1][1] : 34

Enter elements in the second matrix:

element - [0][0] : 45

element - [0][1] : 67

element - [1][0] : 89

element - [1][1] : 11

The First matrix is:

12 18

65 34

The Second matrix is:

45 67

89 11

The multiplication of two matrices is:

2142 1002

5951 4729

**19. Program to Find the Transpose of a Matrix**

#include <stdio.h>

int main()

{

int a[10][10], transpose[10][10], r, c;

printf("Enter rows and columns: ");

scanf("%d %d", &r, &c); // assigning matrix elements printf("\nEnter matrix elements:\n");

for (int i = 0; i< r; ++i)

for (int j = 0; j < c; ++j)

{

printf("Enter element a%d%d: ", i + 1, j + 1);

scanf("%d", &a[i][j]); } // printing the matrix a[][]

printf("\nEntered matrix: \n");

for (int i = 0; i< r; ++i)

for (int j = 0; j < c; ++j)

{

printf("%d ", a[i][j]);

if (j == c - 1)

printf("\n"); } // computing the transpose

for (int i = 0; i< r; ++i)

for (int j = 0; j < c; ++j)

{

transpose[j][i] = a[i][j]; }

transpose printf("\nTranspose of the matrix:\n");

for (int i = 0; i< c; ++i)

for (int j = 0; j < r; ++j)

{

printf("%d ", transpose[i][j]);

if (j == r - 1)

printf("\n");

}

return 0;

}

**OutPut:**

Enter rows and columns: 2 3

Enter element a11: 1

Enter element a12: 4

Enter element a13: 0

Enter element a21: -5

Enter element a22: 2

Enter element a23: 7

Entered matrix:

1 4 0

-5 2 7

Transpose of the matrix:

1 -5

4 2

0 7

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

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int main()

{

char str[100];

int l,i;

printf("\n\nPrinting individual characters of string in reverse order :\n");

printf("------------------------------------------------------\n");

printf("Enter the string:");

fgets(str, sizeof str, stdin);

l=strlen(str);

printf("The characters of the string in reverse are : \n");

for(i=l;i>=0;i--)

{

printf("%c\t", str[i]);

}

printf("\n");

return 0;

}

**OutPut:**

Printing individual characters of string in reverse order :

------------------------------------------------------

Enter the string:nriit college

The characters of the string in reverse are :

e g e l l o c t ii r n

**21. Write a program in C to copy one string to another string.**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int main()

{char str1[100], str2[100];

int i;

printf("\n\nCopy one string into another string:\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("Enter the string:");

fgets(str1, sizeof str1, stdin);

/\* Copies string1 to string2 character by character \*/

i=0;

while(str1[i]!='\0')

{str2[i] = str1[i];

i++;}

//Makes sure that the string is NULL terminated

str2[i] = '\0';

printf("\nThe First string is : %s\n", str1);

printf("The Second string is : %s\n", str2);

printf("Number of characters copied : %d\n", i);

return 0;

}

**OutPut:**

Copy one string into another string:

\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter the string:nriit

The First string is :nriit

The Second string is :nriit

Number of characters copied : 6

**22.To check whether a number is a prime number or not using the function**

#include<stdio.h>

int PrimeOrNot(int);

int main()

{

int n1,prime;

printf("\nFunction to check whether a number is prime number or not\n");

printf("Enter a positive number:");

scanf("%d",&n1);

prime = PrimeOrNot(n1);

if(prime==1)

printf("The number %d is a prime number\n",n1);

else

printf("The number %d is not a prime number\n",n1);

return 0;}

int PrimeOrNot(int n1)

{int i=2;

while(i<=sqrt(n1))

{

if(n1%i==0)

return 0;

else

i++;

}

return 1;

}

**OutPut:**

Function to check whether a number is prime number or not

Enter a positive number:37

The number 37 is a prime number

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

#include <stdio.h>

void swapNumbers(int \*x,int \*y,int \*z);

int main()

{int a,b,c;

printf("\nSwap elements using call by reference\n");

printf("--------------------------------------\n");

printf("\nEnter the value of 1st element:");

scanf("%d",&a);

printf("\nEnter the value of 2nd element:");

scanf("%d",&b);

printf("\nEnter the value of 3rd element:");

scanf("%d",&c);

printf("\nThe values before swapping are:\n");

printf("\nelement1=%d\n element2=%d\n element3=%d\n",a,b,c);

swapNumbers(&a,&b,&c);

printf("\nThe values after swapping are:\n");

printf("element1=%d\n element2=%d\n element3=%d\n\n",a,b,c);

return 0;}

void swapNumbers(int \*x,int \*y,int \*z)

{

int tmp;

tmp=\*y;

\*y=\*x;

\*x=\*z;

\*z=tmp;

}

**OutPut:**

Swap elements using call by reference

--------------------------------------

Enter the value of 1st element:9

Enter the value of 2nd element:18

Enter the value of 3rd element:27

The values before swapping are:

element1=9

element2=18

element3=27

The values after swapping are:

element1=27

element2=9

element3=18

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

#include <stdio.h>

int main()

{

int n1,n2,\*ptr,\*qtr,sum;

printf("\nAdd two numbers using Pointers\n");

printf("\nEnter the first number:");

scanf("%d",&n1);

printf("\nEnter the second number:");

scanf("%d",&n2);

ptr=&n1;

qtr=&n2;

sum=\*ptr+\*qtr;

printf(" The sum of the entered numbers is:%d\n\n",sum);

return 0;

}

**Output:**

Add two numbers using Pointers

Enter the first number:98

Enter the second number:99

The sum of the entered numbers is:197

**25. C program to check whether the given number is palindrome or not.**

#include <stdio.h>

int main()

{

int num,r,sum=0,temp;

printf("Enter a number:");

scanf("%d",&num);

temp=num;

while(num!=0)

{

r=num%10;

sum=sum\*10+r;

num=num/10;

}

if(temp==sum)

printf("%d is a palindrome number\n",temp);

else

printf("%d is not a palindrome number\n",temp);

return 0;

}

**Output:**

Enter a number:121

121 is a palindrome number

Enter a number:2021

2021 is not a palindrome number

**26.C program to calculate GCD(Greatest Common Divisor) of two numbers.**

#include <stdio.h>

int main()

{

int n1,n2,i,gcd;

printf("Enter two integers:");

scanf("%d %d",&n1,&n2);

for(i=1;i<=n1&&i<=n2;++i)

{

// Checks if i is factor of both integers

if(n1%i==0&&n2%i==0)

gcd=i;

}

printf("G.C.D of %d and %d is %d",n1,n2,gcd);

return 0;

}

**Output:**

Enter two integers:5 10

G.C.D of 5 and 10 is 5

**27. Separate odd and even integers in separate arrays**

#include <stdio.h>

void main()

{int arr1[10], arr2[10], arr3[10];

int i,j=0,k=0,n;

printf("\n\nSeparate odd and even integers in separate arrays:\n");

printf("------------------------------------------------------\n");

printf("Input the number of elements to be stored in the array :");

scanf("%d",&n);

printf("Input %d elements in the array :\n",n);

for(i=0;i<n;i++)

{printf("element - %d : ",i);

scanf("%d",&arr1[i]);}

for(i=0;i<n;i++)

{ if (arr1[i]%2 == 0)

{ arr2[j] = arr1[i]; j++;}

else

{arr3[k] = arr1[i];

k++;} }

printf("\nThe Even elements are : \n");

for(i=0;i<j;i++)

{printf("%d ",arr2[i]);}

printf("\nThe Odd elements are :\n");

for(i=0;i<k;i++)

{ printf("%d ", arr3[i]);}

printf("\n\n");

}

**Output:**

Separate odd and even integers in separate arrays:

------------------------------------------------------

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 25

element - 1 : 47

element - 2 : 42

element - 3 : 56

element - 4 : 32

The Even elements are :

42 56 32

The Odd elements are :

25 47

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

#include <stdio.h>

void main()

{ int arr1[100];

int n, i, j, tmp;

printf("\n\nsort elements of array in ascending order :\n ");

printf("Input the size of array : ");

scanf("%d", &n);

printf("Input %d elements in the array :\n",n);

for(i=0;i<n;i++)

{printf("element - %d : ",i);

scanf("%d",&arr1[i]); }

for(i=0; i<n; i++)

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

{if(arr1[j] <arr1[i])

{tmp = arr1[i];

arr1[i] = arr1[j];

arr1[j] = tmp;}}}

printf("\nElements of array in sorted ascending order:\n");

for(i=0; i<n; i++)

{printf("%d ", arr1[i]);}

printf("\n\n");}

**Output:**

sort elements of array in ascending order :

Input the size of array : 5

Input 5 elements in the array :

element - 0 : 2

element - 1 : 7

element - 2 : 4

element - 3 : 5

element - 4 : 9

Elements of array in sorted ascending order:

2 4 5 7 9

**29. Search an element in a row wise and column wise sorted matrix**

#include <stdio.h>

int searchElement(int arr2D[4][4], int n, int x)

{int i = 0, j = n-1;

while ( i< n && j >= 0 )

{if ( arr2D[i][j] == x )

{printf("\nThe element Found at the position in the matrix is: %d, %d", i, j);

return 1; }

if ( arr2D[i][j] > x )

j--;

else

i++;

}

printf("\nThe given element not found in the 2D array.");

return 0; }

int main()

{int arr2D[4][4] = { {15, 23, 31, 39},

{18, 26, 36, 43},

{25, 28, 37, 48},

{30, 34, 39, 50},

};

int i,j,v;

v=37;

printf("The given array in matrix form is : \n");

for(i = 0; i< 4; i++)

{

for (j=0;j<4;j++)

{

printf("%d ", arr2D[i][j]);}

printf("\n");}

printf("The value for searching is: %d",v);

searchElement(arr2D, 4, v);

return 0;}

**Output:**

The given array in matrix form is :

15 23 31 39

18 26 36 43

25 28 37 48

30 34 39 50

The value for searching is: 37

The element Found at the position in the matrix is: 2, 2

**30. Print individual characters of string in reverse order**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

void main()

{

char str[100]; /\* Declares a string of size 100 \*/

int l,i;

printf("\n\nPrint individual characters of string in reverse order :\n");

printf("------------------------------------------------------\n");

printf("Input the string : ");

fgets(str, sizeof str, stdin);

l=strlen(str);

printf("The characters of the string in reverse are : \n");

for(i=l;i>=0;i--)

{

printf("%c ", str[i]);

}

printf("\n");

}

**Output:**

Print individual characters of string in reverse order :

------------------------------------------------------

Input the string : w3resource.com

The characters of the string in reverse are :

m o c . e c r u o s e r 3 w

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

// C program to compare the two strings

// without using strcmp() function

#include <stdio.h>

#definestr\_size 100

int test(char\* s1, char\* s2)

{

int flag = 0;

while (\*s1 != '\0' || \*s2 != '\0') {

if (\*s1 == \*s2) {

s1++;

s2++;

}

else if ((\*s1 == '\0' && \*s2 != '\0')

|| (\*s1 != '\0' && \*s2 == '\0')

|| \*s1 != \*s2) {

flag = 1;

break;

}

}

return flag;}

int main(void)

{char str1[str\_size], str2[str\_size];

int flg=0;

printf("\nInput the 1st string : ");

fgets(str1, sizeof str1, stdin);

printf("Input the 2nd string : ");

fgets(str2, sizeof str2, stdin);

printf("\nString1: %s", str1);

printf("String2: %s", str2);

flg = test(str1, str2);

if(flg == 0)

{

printf("\nStrings are equal.\n");}

else if(flg == 1)

{printf("\nStrings are not equal.");}

return 0;

}

**Output:**

Input the 1st string :aabbcc

Input the 2nd string :abcdef

String1: aabbcc

String2: abcdef

Strings are not equal.

Input the 1st string :abbcdef

Input the 2nd string :abbcdef

String1: abbcdef

String2: abbcdef

Strings are equal.

**32. Write a C Program to Store Information Using Structures with Dynamically Memory**

#include <stdio.h>

#include<stdlib.h>

**struct**course{**int** marks;

**char**subject[30];};

**int**main()

{

**struct** course \*ptr;

**int**i, noOfRecords;

printf("Enter number of records: ");

scanf("%d", &noOfRecords); // Allocates the memory for noOfRecords structures with pointer ptr pointing to the base address.

ptr = (**struct** course\*) malloc (noOfRecords \* **sizeof**(**struct** course));

**for**(i = 0; i<noOfRecords; ++i)

{

printf("Enter name of the subject and marks respectively:\n");

scanf("%s %d", &(ptr+i)->subject, &(ptr+i)->marks); }

printf("Displaying Information:\n");

**for**(i = 0; i<noOfRecords ; ++i)

printf("%s\t%d\n", (ptr+i)->subject, (ptr+i)->marks);

**return** 0;

}

**Output:**

Enter number of records: 2

Enter name of the subject and marks respectively:

RPROGRAMMING 76

Enter name of the subject and marks respectively:

CPROGRAMMING 84

Displaying Information:

RPROGRAMMING76

CPROGRAMMING84

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

#include <stdio.h>

int main()

{

int\* ab;

int m;

m=29;

printf("\n\n Pointer : How to handle the pointers in the program :\n");

printf("------------------------------------------------------------\n");

printf(" Here in the declaration ab = int pointer, int m= 29\n\n");

printf(" Address of m : %p\n",&m);

printf(" Value of m : %d\n\n",m);

ab=&m;

printf(" Now ab is assigned with the address of m.\n");

printf(" Address of pointer ab : %p\n",ab);

printf(" Content of pointer ab : %d\n\n",\*ab);

m=34;

printf(" The value of m assigned to 34 now.\n");

printf(" Address of pointer ab : %p\n",ab);

printf(" Content of pointer ab : %d\n\n",\*ab);

\*ab=7;

printf(" The pointer variable ab is assigned the value 7 now.\n");

printf(" Address of m : %p\n",&m);//as ab contain the address of m

//so \*ab changed the value of m and now m become 7

printf(" Value of m : %d\n\n",m);

return 0;

}

**Output:**

Pointer : How to handle the pointers in the program :

------------------------------------------------------------

Here in the declaration ab = int pointer, int m= 29

Address of m : 0x7ffdd911da8c

Value of m : 29

Now ab is assigned with the address of m.

Address of pointer ab : 0x7ffdd911da8c

Content of pointer ab : 29

The value of m assigned to 34 now.

Address of pointer ab : 0x7ffdd911da8c

Content of pointer ab : 34

The pointer variable ab is assigned the value 7 now.

Address of m : 0x7ffdd911da8c

Value of m : 7

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

#include <stdio.h>

void main()

{int m=300;

float fx = 300.60;

char cht = 'z';

printf("\n\n Pointer : Demonstrate the use of & and \* operator :\n");

printf("--------------------------------------------------------\n");

int \*pt1;

float \*pt2;

char \*pt3;

pt1= &m;

pt2=&fx;

pt3=&cht;

printf( " m = %d\n",m);

printf( "fx = %f\n",fx);

printf( "cht = %c\n",cht);

printf("\n Using & operator :\n");

printf("-----------------------\n");

printf( " address of m = %p\n",&m);

printf( " address of fx = %p\n",&fx);

printf( " address of cht = %p\n",&cht);

printf("\n Using & and \* operator :\n");

printf("-----------------------------\n");

printf( " value at address of m = %d\n",\*(&m));

printf( " value at address of fx = %f\n",\*(&fx));

printf( " value at address of cht = %c\n",\*(&cht));

printf("\n Using only pointer variable :\n");

printf("----------------------------------\n");

printf( " address of m = %p\n",pt1);

printf( " address of fx = %p\n",pt2);

printf( " address of cht = %p\n",pt3);

printf("\n Using only pointer operator :\n");

printf("----------------------------------\n");

printf( " value at address of m = %d\n",\*pt1);

printf( " value at address of fx= %f\n",\*pt2);

printf( " value at address of cht= %c\n\n",\*pt3);

}

**Output:**

Pointer : Demonstrate the use of & and \* operator :

--------------------------------------------------------

m = 300

fx = 300.600006

cht = z

Using &operator :

-----------------------

address of m = 0x7ffeeef55b98

address of fx = 0x7ffeeef55b9c

address of cht = 0x7ffeeef55b97

Using & and \* operator :

-----------------------------

value at address of m = 300

value at address of fx = 300.600006

value at address of cht = z

Using only pointer variable :

----------------------------------

address of m = 0x7ffeeef55b98

address of fx = 0x7ffeeef55b9c

address of cht = 0x7ffeeef55b97

Using only pointer operator :

----------------------------------

value at address of m = 300

value at address of fx= 300.600006

value at address of cht= z

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

#include <stdio.h>

long addTwoNumbers(long \*, long \*);

int main()

{long fno, sno, sum;

printf("\n\n Pointer : Add two numbers using call by reference:\n");

printf("-------------------------------------------------------\n");

printf(" Input the first number : ");

scanf("%ld", &fno);

printf(" Input the second number : ");

scanf("%ld", &sno);

sum = addTwoNumbers(&fno, &sno);

printf(" The sum of %ld and %ld is %ld\n\n", fno, sno, sum);

return 0;}

long addTwoNumbers(long \*n1, long \*n2)

{long sum;

sum = \*n1 + \*n2;

return sum;

}

**Output:**

Pointer : Add two numbers using call by reference:

-------------------------------------------------------

Input the first number : 5

Input the second number : 6

The sum of 5 and 6 is 11

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

#include<stdio.h>

#include<stdlib.h>

int main()

{

int i, num;

float \*data;

printf("Enter elements(1 to 100): ");

scanf("%d",&num);

// allocates the memory for 'num' elements

data=(float\*)calloc(num,sizeof(float));

if(data==NULL)

{

printf("Error! Memory not Allocated.");

exit(0);

}

printf("\n");

// store the number entered by the User

for(i=0;i<num;i++)

{

printf("Enter element %d:",i+1);

scanf("%f",data+i);

}

// store largest number at address data

for(i=0;i<num;i++)

{

if( \*data < \*(data+i))

\*data = \*(data+i);

}

printf("Largest Element = %.2f",\*data);

return 0;

}

**Output:**

Enter elements(1 to 100): 5

Enter element 1:10.2

Enter element 2:23

Enter element 3:23.3

Enter element 4:24

Enter element 5:32

Largest Element = 32.00

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

#include <stdio.h>

int main()

{

char str1[50];

char \*pt;

int ctrV,ctrC;

printf("\n\n Pointer : Count the number of vowels and consonants :\n");

printf("----------------------------------------------------------\n");

printf(" Input a string: ");

fgets(str1, sizeof str1, stdin);

//assign address of str1 to pt

pt=str1;

ctrV=ctrC=0;

while(\*pt!='\0')

{

if(\*pt=='A' ||\*pt=='E' ||\*pt=='I' ||\*pt=='O' ||\*pt=='U' ||\*pt=='a' ||\*pt=='e' ||\*pt=='i' ||\*pt=='o' ||\*pt=='u')

ctrV++;

else

ctrC++;

pt++; //pointer is increasing for searching the next character

}

printf(" Number of vowels : %d\n Number of consonants : %d\n",ctrV,ctrC-1);

return 0;}

**Output:**

Pointer : Count the number of vowels and consonants :

----------------------------------------------------------

Input a string: STRING

Number of vowels : 1

Number of consonants : 5

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

#include <stdio.h>

int\* findLarger(int\*, int\*);

void main()

{int numa=0;

int numb=0;

int \*result;

printf("\n\n Pointer : Show a function returning pointer :\n");

printf(" Input the first number : ");

scanf("%d", &numa);

printf(" Input the second number : ");

scanf("%d", &numb);

result=findLarger(&numa, &numb);

printf(" The number %d is larger. \n\n",\*result);

}

int\* findLarger(int \*n1, int \*n2)

{

if(\*n1 > \*n2)

return n1;

else

return n2;

}

**Output:**

Pointer : Show a function returning pointer :

Input the first number : 5

Input the second number : 6

The number 6 is larger.

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

#*include*<stdio.h>

#*include*<stdlib.h>

*Int* main()

{//1

*Int* i;

*int* count;

*int* \*arr;

*int* sum = 0; //2

printf("Enter number of elements: ");

scanf("%d", &count); //3

arr = (*int*\*)malloc(count \* *sizeof*(*int*));//4

*for* (i = 0; i< count; i++)

{//5

printf("Enter element %d : ", (i + 1));

scanf("%d", arr + i); //6

sum += \*(arr + i);} //7

printf("sum is %d \n", sum); //8

free(arr);

*return* 0;

}

**Output:**

Enter number of elements: 4

Enter element 1 : 10

Enter element 2 : 20

Enter element 3 : 30

Enter element 4 : 40

sum is 100

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

#include<stdio.h>

#include<stdlib.h>

int main(){

int n,i,\*p,sum=0;

printf("Enter number of elements: ");

scanf("%d",&n);

p=(int\*)calloc(n, sizeof(int)); //memory allocated using calloc

if(p == NULL)

{

printf("memory cannot be allocated\n");}

else

{

printf("Enter elements of array:\n");

for(i=0;i<n;++i)

{

scanf("%d",&\*(p+i));

}

for (i = 0; i< n; i++)

{

sum += \*(p + i);}

printf("sum is %d \n", sum);

free(p);

return 0; } }

**Output:**

Enter number of elements: 4

Enter elements of array:

10

20

30

40

sum is 100

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

#include<stdio.h>

long toBin(int);

int main()

{

long bno;

int dno;

printf("\n\n Function : convert decimal to binary :\n");

printf("-------------------------------------------\n");

printf(" Input any decimal number : ");

scanf("%d",&dno);

bno = toBin(dno);

printf("\n The Binary value is : %ld\n\n",bno);

return 0;

}

long toBin(int dno)

{

long bno=0,remainder,f=1;

while(dno != 0)

{remainder = dno % 2;

bno = bno + remainder \* f;

f = f \* 10;

dno = dno / 2;

}

return bno;

}

**Output:**

Function : convert decimal to binary :

-------------------------------------------

Input any decimal number : 65

The Binary value is : 1000001

**42. /\* C program to Convert Decimal to Hexadecimal\*/**

#include <stdio.h>

 int main()

{long decimalnum, quotient, remainder;

int i, j = 0;

char hexadecimalnum[100];

printf("Enter decimal number: ");

scanf("%ld", &decimalnum);

quotient = decimalnum;

while (quotient != 0)

{remainder = quotient % 16;

if (remainder < 10)

hexadecimalnum[j++] = 48 + remainder;

else

hexadecimalnum[j++] = 55 + remainder;

quotient = quotient / 16;

}

*// display integer into character*

for (i = j; i>= 0; i--)

printf("%c", hexadecimalnum[i]);

return 0;

}

**Output:**

Enter decimal number: 62

3E

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

#include<stdio.h>

#define MAX 100

int findMaxElem(int []);

int n;

int main()

{

int arr1[MAX],mxelem,i;

printf("\n\n Function : get largest element of an array :\n");

printf("-------------------------------------------------\n");

printf(" Input the number of elements to be stored in the array :");

scanf("%d",&n);

printf(" Input %d elements in the array :\n",n);

for(i=0;i<n;i++)

{printf(" element - %d : ",i);

scanf("%d",&arr1[i]); }

mxelem=findMaxElem(arr1);

printf(" The largest element in the array is : %d\n\n",mxelem);

return 0;

}

int findMaxElem(int arr1[])

{

int i=1,mxelem;

mxelem=arr1[0];

while(i< n)

{

if(mxelem<arr1[i])

mxelem=arr1[i]; i++;

}

return mxelem;

}

**Output:**

Function : get largest element of an array :

-------------------------------------------------

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

The largest element in the array is : 5

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

#include <stdio.h>

int main ()

{

FILE \* fptr;

int i,n;

char str[100];

char fname[20];

char str1;

printf("\n\n Append multiple lines at end of text file :\n");

printf(" Input the file name to be opened : ");

scanf("%s",fname);

fptr = fopen(fname, "a");

printf(" Input the number of lines to be written : ");

scanf("%d", &n);

printf(" The lines are : \n");

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

{fgets(str, sizeof str, stdin);

fputs(str, fptr);}

fclose (fptr);//Read the file after appended

fptr = fopen (fname, "r");

printf("\n The content of the file %s is :\n",fname);

str1 = fgetc(fptr);

while (str1 != EOF)

{

printf ("%c", str1);

str1 = fgetc(fptr);}

printf("\n\n");

fclose (fptr);

return 0;

}

**Output:**

Append multiple lines at the end of a text Input the file name to be opened : test.txt

Input the number of lines to be written : 3

The lines are :

test line 5

test line 6

test line 7

The content of the file test.txt is :

test line 1

test line 2

test line 3

test line 4

test line 5

test line 6

test line 7

**45. Write a program in C to copy a file in another name**

#include <stdio.h>

#include <stdlib.h>

void main()

{ FILE \*fptr1, \*fptr2;

char ch, fname1[20], fname2[20];

printf("\n\n Copy a file in another name :\n");

printf(" Input the source file name : ");

scanf("%s",fname1);

fptr1=fopen(fname1, "r");

if(fptr1==NULL)

{printf(" File does not found or error in opening.!!");

exit(1);}

printf(" Input the new file name : ");

scanf("%s",fname2);

fptr2=fopen(fname2, "w");

if(fptr2==NULL)

{printf(" File does not found or error in opening.!!");

fclose(fptr1);

exit(2); }

while(1)

{

ch=fgetc(fptr1);

if(ch==EOF)

{break;}

else

{fputc(ch, fptr2);}}

printf(" The file %s copied%s. \n\n",fname1,fname2);

fclose(fptr1);

fclose(fptr2);

getchar();

}

**Output:**

Copy a file in another name :

Input the source file name : test.txt

Input the new file name : test1.txt

The file test.txt copied test1.txt

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

#include <stdio.h>

void main()

{int status;

char fname[20];

printf("\n\n Remove a file from the disk :\n");

printf("----------------------------------\n");

printf(" Input the name of file to delete : ");

scanf("%s",fname);

status=remove(fname);

if(status==0)

{

printf(" The file %s is deleted successfully..!!\n\n",fname);

}

else

{printf(" Unable to delete file %s\n\n",fname);}

}

**Output:**

Remove a file from the disk :

----------------------------------

Input the name of file to delete : test.txt

The file test.txt is deleted successfully..!!

**47. Program to evaluate the following (using loops):**

* + **x-x3/3!+x5/5!-x7/7!+--------------up to n terms**
  + **1+x+x2/2! +x3/3!+--------------up to n terms**
  + **1-x2/2!+x4/4!-x6/6!+------------up to n terms**

# include <stdio.h>

# include <math.h>

Void main()

{ int n, opt, x, f, sign=-1, i;

float sum=0;

printf("Enter number of terms:”);

scanf("%d", &n);

printf("Enter x value: ");

scanf("%d", &x);

printf("1. x - x^3/3! + x^5/5! - x^7/7! + ... \n");

printf("2. 1 + x + x^2/2! + x^3/3! + ... \n");

printf("3. 1 - x^2/2! + x^4/4! - x^6/6! + ... \n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{

case 1:

sum = x;

for(i=3; i<n; i+=2)

{

f \*= (n-1)\*n;

sum += sign\*pow(x,i)/f;

sign \*= -1; }

break;

case 2:

sum = 1;

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

{

f \*= (n-1)\*n;

sum += pow(x,i)/f;

sign \*= -1; }

break;

case 3:

sum = 1;

for(i=2; i<n; i+=2)

{

f \*= (n-1)\*n;

sum += sign\*pow(x,i)/f;

sign \*= -1; }

break; default: printf("Invalid Option ..."); }

printf("Series value: %f\n", sum);

getch();}

**48. to test whether a given number is**

**(using Loops):**

* + **Prime or not**
  + **Perfect or not**
  + **Armstrong or not**
  + **Strong or not**
  + **Palindrome or not**

# include <stdio.h>

# include <math.h>

void main()

{

int opt, num, dnum, i, sum, d, f;

clrscr();

while(1)

{ printf("1. Prime Number \n");

printf("2. Perfect Number \n");

printf("3. Armstrong Number\n");

printf("4. Strong Number\n");

printf("5. Palindrome Number\n");

printf("6. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{

case 1:

printf("Enter a number: ");

scanf("%d", &num);

for(i=2; i<=num/2; i++)

{if(num%i==0)

break; } if(i<=num/2)

printf("%d is Composite\n", num);

else

printf("%d is Prime\n", num);

break;

 case 2:

printf("Enter a number: ");

scanf("%d", &num);

sum=1;

for(i=2; i<=num/2; i++)

{if(num%i==0)

sum+=i;}

if(sum==num)

printf("%d is Perfect\n", num);

else

printf("%d is Not Perfect\n", num);

break;

case 3:

printf("Enter a number: ");

scanf("%d", &num);

sum = 0;

dnum=num;

while(dnum)

{d = dnum%10;

sum += d\*d\*d;

dnum/=10;

} if(sum==num)

printf("% Armstrong\n", num);

else

printf("%d is Not Armstrong\n", num);

break;

case 4:

printf("Enter a number: ");

scanf("%d", &num);

sum = 0;

dnum=num;

while(dnum)

{d = dnum%10;

for(i=2, f=1; i<=d; i++)

f\*=i;

sum += f;

dnum/=10;

} if(sum==num)

printf("%d is Strong\n", num);

else

printf("%d is Not Strong\n", num);

break;

case 5:

printf("Enter a number: ");

scanf("%d", &num);

sum = 0;

dnum=num;

while(dnum)

{sum = sum\*10+dnum%10;

dnum/=10; }

if(sum==num)

printf("%d is Palindrome\n", num);

else

printf("%d is Not Palindrome\n", num);

break;

case 6: return 0;

default:

printf("Invalid Option...");}

}

}

**Output:**

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 1

Enter a number: 17

17 is Prime

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 1

Enter a number: 12

12 is Composite

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 2

Enter a number: 6

6 is Perfect

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 2

Enter a number: 8

8 is Not Perfect

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 3

Enter a number: 153

153 Armstrong

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 3

Enter a number: 135

135 is Not Armstrong

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 4

Enter a number: 145

145 is Strong

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 4

Enter a number: 154

154 is Not Strong

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 5

Enter a number: 121

121 is Palindrome

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 5

Enter a number: 122

122 is Not Palindrome

1. Prime Number

2. Perfect Number

3. Armstrong Number

4. Strong Number

5. Palindrome Number

6. Exit

Enter your option: 6

**49. display statistical parameters (using one - dimensional array)**

* + - **Mean**
    - **Median**
    - **Mode**

**Standard deviation**

# include <stdio.h>

# include <math.h>

float getMean(float a[], int n)

{

float sum=0;

int i;

for(i=0; i<n; i++)

sum+=a[i];

return sum; }

void sort(float a[], int n)

{ int i, j, temp;

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; } }

float getMode(float a[], int n)

{ floatmaxValue = 0;

int maxCount = 0, i, j;

sort(a, n);

for(i=0; i<n; i++)

printf("%f ", a[i]);

for (i = 0; i< n; ++i)

{ int count = 0;

for (j = 0; j < n; ++j)

{ if (a[j] == a[i])

++count; }

if (count >maxCount)

{ maxCount = count;

maxValue = a[i]; } }

return maxValue;

}

float getMedian(float a[], int n)

{ int x;

x = n/2;

sort(a, n);

return (n%2!=0)?a[x]:(a[x-1]+a[x])/2;

}

float getSD(float a[], int n)

{ float x, sum;

int i;

x = getMean(a, n);

for(i=0; i<n; i++)

sum += (x-a[i])\*(x-a[i]);

return sqrt(sum/n); }

main()

{ int n, i, opt;

float \*p, x;

printf("Enter number of elements: ");

scanf("%d", &n);

p = (float\*)malloc(n\*sizeof(float));

printf("Enter %d elements: ", n);

for(i=0; i<n; i++)

scanf("%f", &p[i]);

while(1)

{

printf("1. Mean \n");

printf("2. Median \n");

printf("3. Mode\n");

printf("4. Standard Deviation\n");

printf("5. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{ case 1: x = getMean(p, n);

printf("\nMean : %f\n", x/n); break;

case 2: x = getMedian(p, n);

printf("\nMedian: %f\n", x); break;

case 3: x = getMode(p, n);

printf("\nMode: %f\n", x); break;

case 4: x = getSD(p, n);

printf("\nStandard Deviation: %f\n", x); break;

case 5: return 0;

default: printf("Invalid Option ..."); }}}

**Output:**

Enter number of elements: 3

Enter 3 elements: 10

20

30

1. Mean

2. Median

3. Mode

4. Standard Deviation

5. Exit

Enter your option: 1

Mean : 20.000000

1. Mean

2. Median

3. Mode

4. Standard Deviation

5. Exit

Enter your option: 2

Median: 20.000000

1. Mean

2. Median

3. Mode

4. Standard Deviation

5. Exit

Enter your option: 3

10.000000 20.000000 30.000000

Mode: 10.000000

1. Mean

2. Median

3. Mode

4. Standard Deviation

5. Exit

Enter your option: 4

Standard Deviation: 40.8248291. Mean

2. Median

3. Mode

4. Standard Deviation

5. Exit

Enter your option: 5

**50. to perform the following operations in a list (using one -Dimensional array)**

* + - **Insertion of an element**
    - **Deletion of an element**
    - **Remove duplicates form the list**

# include <stdio.h>

# include <math.h>

main()

{ int x, i,j, k, opt;

int a[100], p=-1;

while(1)

{ printf("\n1. Insertion \n");

printf("2. Deletion \n");

printf("3. Remove Duplicates\n");

printf("4. Display\n");

printf("5. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

if(p<99)

{ printf("Enter a number: ");

scanf("%d", &a[++p]);}

else

printf("Array is Full");

break;

case 2:

printf("Enter element to delete: ");

scanf("%d", &x);

for(i=0; i<=p; i++)

if(a[i]==x)

{ a[i] = a[p--];

break; }

if(i>p)

printf("Element not found");

break;

case 3:

for(i=0; i<p; i++)

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

if(a[i]==a[j])

{for(k=j; k<=p; k++)

a[k]=a[k+1];

p--;

j--; }

break;

case 4:

for(i=0; i<=p; i++)

printf("%d ", a[i]);

break;

case 5: return 0;

default: printf("Invalid Option ...");

} } }

**Output:**

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 1

Enter a number: 20

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

20

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 1

Enter a number: 30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

20 30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 2

Enter element to delete: 40

Element not found

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 2

Enter element to delete: 20

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 1

Enter a number: 30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

30 30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 3

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 3

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 4

30

1. Insertion

2. Deletion

3. Remove Duplicates

4. Display

5. Exit

Enter your option: 5

**51.A menu driven program with options (using two dimensional arrays)**

* 1. **To compute A+B**
  2. **To compute A x B**
  3. **To find transpose of matrix A.**

**Where A and B are matrices**.

# include <stdio.h>

# include <math.h>

 void matrixAdd()

{int m1, n1, m2, n2, i, j;

int a[10][10], b[10][10], c[10][10];

printf("Enter first matrix size: ");

scanf("%d %d", &m1, &n1);

printf("Enter second matrix size: ");

scanf("%d %d", &m2, &n2);

if(m1!=m2 || n1!=n2)

{ printf("Matrix Addition is not possible");

return; }

printf("Enter first matrix elements (%dx%d):\n", m1, n1);

for(i=0; i<m1; i++)

for(j=0; j<n1; j++)

scanf("%d", &a[i][j]);

printf("Enter second matrix elements (%dx%d):\n", m2, n2);

for(i=0; i<m2; i++)

for(j=0; j<n2; j++)

scanf("%d", &b[i][j]);

for(i=0; i<m2; i++)

for(j=0; j<n2; j++)

c[i][j]=a[i][j]+b[i][j];

printf("Output Matrix (%dx%d): \n", m1,n1);

for(i=0; i<m2; i++)

{ for(j=0; j<n2; j++)

printf("%d ", c[i][j]);

printf("\n"); }

}

void matrixMul()

{ int m1, n1, m2, n2, i, j, k;

int a[10][10], b[10][10], c[10][10]={0};

printf("Enter first matrix size: ");

scanf("%d %d", &m1, &n1);

printf("Enter second matrix size: ");

scanf("%d %d", &m2, &n2);

if(n1 != m2)

{ printf("Matrix Multiplication is not possible");

return; }

printf("Enter first matrix elements (%dx%d):\n", m1, n1);

for(i=0; i<m1; i++)

for(j=0; j<n1; j++)

scanf("%d", &a[i][j]);

printf("Enter second matrix elements (%dx%d):\n", m2, n2);

for(i=0; i<m2; i++)

for(j=0; j<n2; j++)

scanf("%d", &b[i][j]);

for(i=0; i<m1; i++)

for(j=0; j<n2; j++)

for(k=0;k<n1; k++)

c[i][j]+=a[i][k]\*b[k][j];

printf("Output Matrix (%dx%d): \n", m1,n2);

for(i=0; i<m2; i++)

{ for(j=0; j<n2; j++)

printf("%d ", c[i][j]);

printf("\n");

} }

 void matrixTrans()

{int m, n, i, j;

int a[10][10], t[10][10];

printf("Enter first matrix size: ");

scanf("%d %d", &m, &n);

printf("Enter elements (%dx%d):\n",m,n);

for(i=0; i<m; i++)

for(j=0; j<n; j++)

scanf("%d", &a[i][j]);

for(i=0; i<m; i++)

for(j=0; j<n; j++)

t[j][i]=a[i][j];

printf("Transposed Matrix (%dx%d): \n", n, m);

for(i=0; i<n; i++)

{ for(j=0; j<m; j++)

printf("%d ", t[i][j]);

printf("\n"); } }

 main()

{ int opt;

while(1)

{printf("\n1. Matrix Addition\n");

printf("2. Matrix Multiplication \n");

printf("3. Matrix Transpose \n");

printf("4. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt); switch(opt)

{case 1: matrixAdd(); break;

case 2: matrixMul(); break;

case 3: matrixTrans(); break;

case 4: return 0;

default: printf("Invalid Option ...");

}} }

**Output:**

1. Matrix Addition

2. Matrix Multiplication

3. Matrix Transpose

4. Exit

Enter your option: 1

Enter first matrix size: 2 2

Enter second matrix size: 2 2

Enter first matrix elements (2x2):

1

2

3

4

Enter second matrix elements (2x2):

5

6

7

8

Output Matrix (2x2):

6 8

10 12

1. Matrix Addition

2. Matrix Multiplication

3. Matrix Transpose

4. Exit

Enter your option: 2

Enter first matrix size: 2

2

Enter second matrix size: 2 2

Enter first matrix elements (2x2):

2

1

0

3

Enter second matrix elements (2x2):

4

2

1

5

Output Matrix (2x2):

9 9

3 15

1. Matrix Addition

2. Matrix Multiplication

3. Matrix Transpose

4. Exit

Enter your option: 3

Enter first matrix size: 2 2

Enter elements (2x2):

1

2

3

4

Transposed Matrix (2x2):

1 3

2 4

1. Matrix Addition

2. Matrix Multiplication

3. Matrix Transpose

4. Exit

Enter your option: 4

**52. to perform the following using Strings**

**To test the given string is palindrome or not**

**To sort strings in alphabetical order**

# include <stdio.h>

# include <string.h>

main()

{ int i, j, opt, n, l2, l, f;

char names[50][50], x[50];

while(1)

{ printf("\n1. Palindrome \n");

printf("2. Sorting names \n");

printf("3. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

printf("Enter a string: ");

scanf("%s", x);

l = strlen(x);

l2 = l/2;

for(f=1, i=0; i<l2; i++)

if(x[i] != x[l-i-1])

{ f=0;

break; }

 if(f==0)

printf("%s is Not Palindrome", x);

else

printf("%s is Palindrome", x);

break;

case 2:

 printf("Enter number of names: ");

scanf("%d", &n);

printf("Enter names: ");

for(i=0; i<n; i++)

scanf("%s", names[i]);

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

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

if(strcmp(names[i],names[j])>0)

{

strcpy(x, names[i]);

strcpy(names[i], names[j]);

strcpy(names[j],x); }

printf("Names in Sorted Order: \n");

for(i=0; i<n; i++)

printf("%s \n", names[i]);

break;

case 3:

return 0;

default: printf("Invalid Option ...");

}} }

**Output:**

1. Palindrome

2. Sorting names

3. Exit

Enter your option: 1

Enter a string: madam

madam is Palindrome

1. Palindrome

2. Sorting names

3. Exit

Enter your option: 1

Enter a string: madan

madan is Not Palindrome

1. Palindrome

2. Sorting names

3. Exit

Enter your option: 2

Enter number of names: 4

Enter names: sri

srinivasarao

srinu

Names in Sorted Order:

rao

sri

srinivasa

srinu

1. Palindrome

2. Sorting names

3. Exit

Enter your option: 3

**53. To find the Factorial value**

**To generate Fibonacci series**

**To find the GCD of two given numbers**

# include <stdio.h>

int factorial(int n)

{

if(n==1||n==0)

return 1;

return n\*factorial(n-1);

}

int fibonacci(int n)

{

if(n==0||n==1)

return n;

return fibonacci(n-1)+fibonacci(n-2);

}

int gcd(int n1, int n2)

{

if(n2==0)

return n1;

return gcd(n2, n1%n2);

}

main()

{

int n1, n2, i, x, opt;

while(1)

{

printf("\n1. Factorial \n");

printf("2. Fibonacci \n");

printf("3. GCD \n");

printf("4. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

printf("Enter a number : ");

scanf("%d", &n1);

x = factorial(n1);

printf("\nFactorial: %d", x);

break;

case 2:

printf("Enter number of elements : ");

scanf("%d", &n1);

for(i=0; i<n1; i++)

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

break;

case 3:

printf("Enter 2 numbers: ");

scanf("%d %d", &n1, &n2);

x = gcd(n1, n2);

printf("gcd: %d ", x);

break;

case 4:

return 0;

default: printf("Invalid Option ...");

}} }

**Output:**

1. Factorial

2. Fibonacci

3. GCD

4. Exit

Enter your option: 1

Enter a number: 4

Factorial: 24

1. Factorial

2. Fibonacci

3. GCD

4. Exit

Enter your option: 2

Enter number of elements: 4

0 1 1 2

1. Factorial

2. Fibonacci

3. GCD

4. Exit

Enter your option: 3

Enter 2 numbers: 24 12

gcd: 12

1. Factorial

2. Fibonacci

3. GCD

4. Exit

Enter your option: 4

**54. A menu driven program with options (using dynamic memory allocation)**

**Linear search**

**Binary search**

# include <stdio.h>

int linearSearch(int a[], int n, int x)

{ inti;

for(i=0; i<n; i++)

if(a[i]==x)

return 1;

return 0; }

void sort(int a[], int n)

{ inti, j, temp;

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;

} }

int binarySearch(int a[], int n, int x)

{ int l, r, mid;

l = 0; r = n-1;

while(l<r)

{ mid = (l+r)/2;

if(a[mid]==x)

return 1;

else if(a[mid]>x)

r = mid-1;

else

l = mid+1; }

return 0;

}

main()

{int a[100], i, opt, se, n;

clrscr();

while(1)

{ printf("\n1. Linear Search \n");

printf("2. Binary Search \n");

printf("3. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

printf("Enter number of elements : ");

scanf("%d", &n);

printf("Enter %d elements : \n", n);

for(i=0; i<n; i++)

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

printf("Enter Search element : ");

scanf("%d", &se);

if(linearSearch(a, n, se))

printf("%d is Found", se);

else

printf("%d is not Found", se);

break;

case 2:

printf("Enter number of elements : ");

scanf("%d", &n);

printf("Enter %d elements : \n", n);

for(i=0; i<n; i++)

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

printf("Enter Search element : ");

scanf("%d", &se);

sort(a, n);

if(binarySearch(a, n, se))

printf("%d is Found", se);

else

printf("%d is not Found", se);

break;

 case 3: return 0;

 default: printf("Invalid Option ...");

}

}

}

**Output:**

1. Linear Search

2. Binary Search

3. Exit

Enter your option: 1

Enter number of elements : 4

Enter 4 elements :

10

20

40

30

Enter Search element : 40

40 is Found

1. Linear Search

2. Binary Search

3. Exit

Enter your option: 2

Enter number of elements : 10

Enter 10 elements :

20

40

30

60

80

90

45

53

23

34

Enter Search element : 45

45 is Found

1. Linear Search

2. Binary Search

3. Exit

Enter your option: 3

**55. A menu driven program with options (using Character array of pointers)**

* + - **To insert a student name**
    - **To delete a name**
    - **To sort names in alphabetical order**
    - **To print list of names**

# include <stdio.h>

# include <math.h>

main()

{int i, j, opt, p=-1;

char names[50][50], x[50];

while(1)

{printf("\n1. Insertion \n");

printf("2. Deletion \n");

printf("3. Sort names\n");

printf("4. Display\n");

printf("5. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

if(p<99)

{

printf("Enter a name: ");

scanf("%s", names[++p]); }

else

printf("Array is Full");

break;

case 2:

printf("Enter a string to delete: ");

scanf("%s", x);

for(i=0; i<=p; i++)

if(strcmp(names[i],x)==0)

{strcpy(names[i], names[p--]);

break; }

if(i>p)

printf("String not found");

break;

case 3:

for(i=0; i<p; i++)

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

if(strcmp(names[i],names[j])>0)

{

strcpy(x, names[i]);

strcpy(names[i], names[j]);

strcpy(names[j],x);

}

break;

case 4:

for(i=0; i<=p; i++)

printf("\n%s ", names[i]);

break;

case 5: return 0;

default: printf("Invalid Option ...");

}

}

}

**Output:**

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 1

Enter a name: srinu

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 1

Enter a name: sravanthi

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 1

Enter a name: pavithra

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 1

Enter a name: jaiveer

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 3

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 4

jaiveer

pavithra

sravanthi

srinu

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 2

Enter a string to delete: jaiveer

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 4

srinu

pavithra

sravanthi

1.Insertion

2. Deletion

3. Sort names

4. Display

5. Exit

Enter your option: 5

1. **Write a program to perform the following operations on Complex numbers (using Structures & pointers):** 
   1. **Read a Complex number**
   2. **Addition, subtraction and multiplication of two complex numbers**
   3. **Display a Complex number**

# include <stdio.h>

struct complex

{

int rel;

int img; };

void readComplex(struct complex \*c)

{printf("Enter real and imaginary parts: ");

scanf("%d%d", &c->rel, &c->img); }

void addComplex(struct complex c1, struct complex c2, struct complex \*c3)

{c3->rel = c1.rel + c2.rel;

c3->img = c1.img + c2.img; }

void subComplex(struct complex c1, struct complex c2, struct complex \*c3)

{c3->rel = c1.rel - c2.rel;

c3->img = c1.img - c2.img;

}

 void mulComplex(struct complex c1, struct complex c2, struct complex \*c3)

{c3->rel = c1.rel\*c2.rel-c1.img\*c2.img;

c3->img = c1.rel\*c2.img+c1.img\*c2.rel; }

 void dispComplex(struct complex c)

{ printf("\n%d", c.rel);

if(c.img<0)

printf("-i%d\n", -1\*c.img);

else

printf("+i%d\n", c.img); }

main()

{struct complex c1, c2, c3;

int opt;

while(1)

{printf("\n1. Read 2 Complex numbers\n");

printf("2. Addition \n");

printf("3. Subtraction \n");

printf("4. Multiplication\n");

printf("5. Exit\n");

printf("Enter your option: ");

scanf("%d", &opt);

switch(opt)

{case 1:

readComplex(&c1);

readComplex(&c2);

break;

case 2:

addComplex(c1, c2, &c3);

dispComplex(c3);

break;

case 3:

subComplex(c1, c2, &c3);

dispComplex(c3);

break;

case 4:

mulComplex(c1, c2, &c3);

dispComplex(c3);

break;

case 5:

return 0;

default: printf("Invalid Option ...");

} } }

**Output:**

1. Read 2 Complex numbers

2. Addition

3. Subtraction

4. Multiplication

5. Exit

Enter your option: 1

Enter real and imaginary parts: 2 3

Enter real and imaginary parts: 2 3

1. Read 2 Complex numbers

2. Addition

3. Subtraction

4. Multiplication

5. Exit

Enter your option: 2

4+i6

1. Read 2 Complex numbers

2. Addition

3. Subtraction

4. Multiplication

5. Exit

Enter your option: 3

0+i0

1. Read 2 Complex numbers

2. Addition

3. Subtraction

4. Multiplication

5. Exit

Enter your option: 4

-5+i12

1. Read 2 Complex numbers

2. Addition

3. Subtraction

4. Multiplication

5. Exit

Enter your option: 5

1. **Write a program to find out day for given date**

// Calender Program

#include<stdio.h>

#include<stdlib.h>

void main()

{int dd,mm,yy,oddyy,oddy1,yy1,yy2,yy3,yy4,y,y1,week,m;

printf("\n Enter Date :");

scanf("%d",&dd);

printf("\n Enter Month :");

scanf("%d",&mm);

printf("\n Enter Year :");

scanf("%d",&yy);

if((mm < 1) || (mm > 12))

    {printf("\n Invalid Date");

        exit(0);}

if((mm > 0) || (mm < 12))

{if((mm==1) || (mm==3) || (mm==5) || (mm==7) || (mm==8) || (mm==10) || (mm==12))

{   if((dd < 1) || (dd > 31))

    {       printf("\n Invalid Date");

        exit(0);} }

else if((mm==4) || (mm==6) || (mm==9)|| (mm==11))

{if((dd < 1) || (dd > 30))

    {       printf("\n Invalid Date");

        exit(0);}}

else if(mm==2)

{if(yy %4 == 0 && yy%400 !=100 && yy%400 !=200 && yy%400 !=300)

    {       if((dd < 1) || (dd > 29))

        {       printf("\n Invalid Date");

            exit(0);}}

    else

    {   if((dd < 1) || (dd > 28))

        {       printf("\n Invalid Date");

            exit(0);}}}}

else

{printf("\n Invalid Date");

    exit(0);}

printf("\n Given Date is %d - %d - %d of ",dd,mm,yy);

yy1=(yy-1)%400;

if((yy1>=0) && (yy1<100))

    oddyy=0;

else if((yy1>=100) &&(yy1 < 200))

    oddyy=5;

else if((yy1 >= 200) && (yy1 < 300))

    oddyy=3;

else if ((yy1 >= 300) && (yy1 < 400))

    oddyy=1;

yy2=(yy-1)%100;

yy3=yy2/4;

yy4=(yy2+1)%4;

oddy1=yy2+yy3;

if(mm==1) m=0;

if(mm==2) m=3;

if(yy4!=0 ||  yy%400 ==100 ||  yy%400 ==200 ||  yy%400 ==300)

{if(mm==3) m=3;

if(mm==4) m=6;

if(mm==5) m=1;

if(mm==6) m=4;

if(mm==7) m=6;

if(mm==8) m=2;

if(mm==9) m=5;

if(mm==10) m=0;

if(mm==11) m=3;

if(mm==12) m=5;}

else

{if(mm==3) m=4;

if(mm==4) m=0;

if(mm==5) m=2;

if(mm==6) m=5;

if(mm==7) m=0;

if(mm==8) m=3;

if(mm==9) m=6;

if(mm==10) m=1;

if(mm==11) m=4;

if(mm==12) m=6;}

week=oddy1+dd+m+oddyy;

week=week%7;

if(week == 0)

    printf("Sunday");

if(week == 1)

    printf("Monday");

if(week == 2)

    printf(" Tuesday");

if(week == 3)

    printf("Wedensday");

if(week == 4)

    printf("Thursday");

if(week == 5)

    printf("Friday");

if(week == 6)

    printf("Saturday");

}

1. List out Prime numbers between a given interval

#include <stdio.h>

  #include <math.h>

 main()

{  int a,b,i,j,s1,flag;

    printf("Enter lower bound of the interval: ");

    scanf("%d", &a);

    printf("\n Enter upper bound of the interval: ");

    scanf("%d", &b);

    printf("\nPrime numbers between %d and %d are: ", a, b);

    for (i = a; i <= b; i++)

     {

if (i == 1 || i == 0)

            continue;

flag = 1;

s1=floor(sqrt(i));

        for (j = 2; j <= s1; ++j) {

            if (i % j == 0) {

                flag = 0;

                break;  }}

        // flag = 1 means i is prime

        if (flag == 1)

            printf("%d ", i); }

}

1. Conversion of binary to decimal number

#include<stdio.h>

main()

{

int dno,bno=0,remainder,f=1;;

printf("\n\n Function : convert binary to decimal :\n");

printf("-------------------------------------------\n");

printf(" Input any binary  number : ");

scanf("%d",&bno);

while(bno != 0)

    {remainder = bno % 10;

dno = dno + remainder \* f;

         f = f \* 2;

bno = bno /10; }

printf("\n Thedecimal value is : %d\n",dno);}

1. Conversion of decimal to binary number

#include<stdio.h>

long toBin(int);

main()

{

long bno;

int dno;

printf("\n\n Function : convert decimal to binary :\n");

printf("-------------------------------------------\n");

printf(" Input any decimal number : ");

scanf("%d",&dno);

bno = toBin(dno);

printf("\n The Binary value is : %ld\n\n",bno);

return 0;

}

long toBin(int dno)

{long bno=0,remainder,f=1;

while(dno != 0)

    {remainder = dno % 2;

bno = bno + remainder \* f;

         f = f \* 10;

dno = dno / 2; }

return bno;}

1. Conversion of decimal to octal number

#include<stdio.h>

main()

{

int dno, ono, remainder,f=1;;

printf("\n\n Function : convert decimal to octal :\n");

printf("-------------------------------------------\n");

printf(" Input any decimal number : ");

scanf("%d",&dno);

while(dno != 0)

    {remainder = dno % 8;

ono = ono + remainder \* f;

         f = f \* 10;

dno = dno / 8; }

printf("\n The Octal value is : %d\n",ono);}

1. Conversion of octal to decimal number

#include<stdio.h>

main()

{

int dno,ono=0,remainder,f=1;;

printf("\n\n Function : convert octal to decimal :\n");

printf("-------------------------------------------\n");

printf(" Input any octal number : ");

scanf("%d",&ono);

while(ono != 0)

    {remainder = ono % 10;

dno = dno + remainder \* f;

         f = f \* 8;

ono = ono /10; }

printf("\n The decimal value is : %d\n",dno);}

1. List of natural numbers

#include<stdio.h>

int main()

{

    int i,n;

    printf("\n Enter n value");

    scanf("%d",&n);

    for(i=1;i<=n;i++)

        printf("%d ",i);

}

1. Calculation HCF or GCD

#include <stdio.h>

main()

{int n1,n2,hcf,mini;

printf("Enter two integers:");

scanf("%d %d",&n1,&n2);

if(n1>n2)

{mini=n2;}

else

{mini=n1;}

while(n1%mini!=0 || n2%mini!=0)

{mini=mini-1;

}

hcf=mini;

printf("HCF of %d and %d is %d",n1,n2,hcf);}

1. Calculation HCF or GCD

// Calender Program

#include <stdio.h>

main()

{int n1,n2,i,gcd;

printf("Enter two integers:");

scanf("%d %d",&n1,&n2);

for(i=1;i<=n1&&i<=n2;++i)

    { // Checks if i is factor of both integers

if(n1%i==0&&n2%i==0)

gcd=i;}

printf("G.C.D of %d and %d is %d",n1,n2,gcd);}

1. Conversion of hexadecimal to decimal number

#include <stdio.h>

#include <math.h>

#include <string.h>

#define ARRAY\_SIZE 20

main()

{ char hex[ARRAY\_SIZE];

long long decimal = 0, base = 1;

int i = 0, value, length;

/\* Get hexadecimal value from user \*/

printf("Enter hexadecimal number: ");

fflush(stdin);

fgets(hex,ARRAY\_SIZE,stdin);

length = strlen(hex);

for(i = length--; i >= 0; i--)

{if(hex[i] >= '0' && hex[i] <= '9')

{decimal += (hex[i] - 48) \* base;

base \*= 16;}

else if(hex[i] >= 'A' && hex[i] <= 'F')

{decimal += (hex[i] - 55) \* base;

base \*= 16;}

else if(hex[i] >= 'a' && hex[i] <= 'f')

{decimal += (hex[i] - 87) \* base;

base \*= 16;} }

printf("\nHexadecimal number = %s", hex);

printf("Decimal number = %lld\n", decimal);}

1. Calculation LCM

#include <stdio.h>

main()

{int n1,n2,lcm,maxi;

printf("Enter two integers:");

scanf("%d %d",&n1,&n2);

if(n1>n2)

{maxi=n1;}

else

{maxi=n2;}

while(maxi%n1!=0 || maxi%n2!=0)

{maxi=maxi+1;

}

lcm=maxi;

printf("LCM of %d and %d is %d",n1,n2,lcm);}

1. Calculation LCM

// Calender Program

#include <stdio.h>

main()

{int n1,n2,i,lcm, maxi;

printf("Enter two integers:");

scanf("%d %d",&n1,&n2);

if(n1>n2)

{maxi=n1;}

else

{maxi=n2;}

for(i=maxi;i<=n1\*n2;i++)

    { // Checks if i is factor of both integers

if(i%n1==0 && i%n2==0)

break;}

lcm=i;

printf("LCM of %d and %d is %d",n1,n2,lcm);}

1. hello

#include<stdio.h>

void main()

{

    printf("hello");

}