1. **Switch Case**

#include <stdio.h>

void main ()

{

int choice,r,l,w,b,h;

float area;

printf("Input 1 for area of circle\n");

printf("Input 2 for area of rectangle\n");

printf("Input 3 for area of triangle\n");

printf("Input your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("Input radious of the circle : ");

scanf("%d",&r);

area=3.14\*r\*r;

break;

case 2:

printf("Input length and width of the rectangle : ");

scanf("%d%d",&l,&w);

area=l\*w;

break;

case 3:

printf("Input the base and hight of the triangle :");

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

area=.5\*b\*h;

break; }

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

}

**Code in Array**

**2. Grading\_system\_withcase\_statement**

#include<stdio.h>

int main(){

int marks ,grade;

printf("Input the numbers \n");

scanf("%d",&marks);

grade=marks/10;

switch(grade)

{ case 10:

case 9:

printf("Your grade is A \n");

break;

case 8:

printf("Your grade is B \n");

break;

case 7:

printf("Your grade is C \n");

break;

case 6:

printf("Your grade is D \n");

break;

default:

printf("Your grade is F \n");

break;

}

}

**3. Idendity\_matrix I2​=[1 0​\_01​] I3​=⎣​100\_01​0\_001​⎦**

#include <stdio.h>

//In a square matrix if all the main diagonal elements are 1's and

//all the remaining elements are 0's is called an Identity Matrix.

void main()

{

int arr1[10][10];

int r1,c1;

int i, j, yn =1;

printf("\n\n Check whether a given matrix is an identity matrix :\n ");

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

printf("Input number of Rows for the matrix :");

scanf("%d", &r1);

printf("Input number of Columns for the matrix :");

scanf("%d",&c1);

printf("Input 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",&arr1[i][j]);

}

}

printf("The matrix is :\n");

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

{

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

printf("% 4d",arr1[i][j]);

printf("\n");

}

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

{

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

{

if(arr1[i][j] != 1 && arr1[j][i] !=0)

{

yn = 0;

break;

}

} }

if(yn == 1 )

printf(" The matrix is an identity matrix.\n\n");

else

printf(" The matrix is not an identity matrix.\n\n");

}

**4. insert new element array**

#include <stdio.h>

void main()

{

int arr1[100],i,n,p,inval;

printf("\n\nInsert New value in the sorted array :\n");

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

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

scanf("%d", &n);

/\* Stored values into the array\*/

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

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

{

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

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

}

printf("Input the value to be inserted : ");

scanf("%d",&inval);

printf("The exist array list is :\n ");

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

printf("% 5d",arr1[i]);

/\* Determine the position where the new value will be insert.\*/

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

if(inval<arr1[i])

{

p = i;

break;

}

/\* move all data at right side of the array \*/

for(i=n;i>=p;i--)

arr1[i]= arr1[i-1];

/\* insert value at the proper position \*/

arr1[p]=inval;

printf("\n\nAfter Insert the list is :\n ");

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

printf("% 5d",arr1[i]);

printf("\n");

}

**5. Matrix multiplication**

#include <stdio.h>

void main()

{

int arr1[50][50],brr1[50][50],crr1[50][50],i,j,k,r1,c1,r2,c2,sum=0;

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

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

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

printf("\nInput 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("Input 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",&arr1[i][j]);

}

}

printf("Input 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",&brr1[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",arr1[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",brr1[i][j]);

}

//multiplication of matrix

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

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

crr1[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+arr1[i][k]\*brr1[k][j];

crr1[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",crr1[i][j]);

}

}

}

printf("\n\n");

}

**6.diagonal matrix (1 2 3\_3 2 5\_4 3 1) diagonal is 1+2+1=4**

#include <stdio.h>

void main()

{ int i,j,arr1[50][50],sum=0,n;

printf("\n\nFind sum of right diagonals of a matrix :\n");

printf("Input the size of the square matrix : ");

scanf("%d", &n);

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

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

{

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

{

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

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

if (i==j) sum= sum+arr1[i][j];

}

}

printf("The matrix is :\n");

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

{

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

printf("% 4d",arr1[i][j]);

printf("\n");

}

printf("Addition of the right Diagonal elements is :%d\n",sum);

}

**7. total duplicate elements in array**

#include <stdio.h>

void main()

{

int arr1[100];

int arr2[100];

int arr3[100];

int n,mm=1,ctr=0;

int i, j;

printf("\n\nCount total number of duplicate elements in an array:\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++)

{

arr2[i]=arr1[i];

arr3[i]=0;

}

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

{

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

{

if(arr1[i]==arr2[j])

{

arr3[j]=mm;

mm++;

}

}

mm=1;

}

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

{

if(arr3[i]==2){ctr++;}

}

printf("The total number of duplicate elements found in the array is: %d \n", ctr);

printf("\n\n");

}

**8. transpose matrix (1 2 3\_4 5 6\_7 8 9)->(1 4 7\_2 5 8\_3 6 9)**

#include <stdio.h>

void main()

{

int arr1[50][50],brr1[50][50],i,j,k=0,r,c;

printf("\n\nTranspose of a Matrix :\n");

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

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

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

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

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

{

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

{

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

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

}

}

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

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

{

printf("\n");

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

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

}

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

{

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

{

brr1[j][i]=arr1[i][j];

} }

printf("\n\nThe transpose of a matrix is : ");

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

printf("\n");

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

printf("%d\t",brr1[i][j]);

}

}

printf("\n\n");

}

**9. array input into output**

#include<stdio.h>

int main()

{

int a[10],i,j,n;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

printf("output:\n");

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

}

**10. array sum**

#include<stdio.h>

int main()

{

int a[10],i,j,n,sum=0;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

printf("output:\n");

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

sum=sum+a[i];

printf("%d ",sum);

}

**11. ascending array**

#include<stdio.h>

int main()

{

int a[100],i,j,n,k,tmp;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

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

{

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

{

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

{

tmp = a[i];

a[i] = a[j];

a[j] = tmp;

}

}

}

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

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

{

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

}

printf("\n\n");

}

**12. decending array**

#include<stdio.h>

int main()

{

int a[100],i,j,n,k,tmp;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

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

{

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

{

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

{

tmp = a[i];

a[i] = a[j];

a[j] = tmp;

}

}

}

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

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

{

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

}

printf("\n\n");

}

**13. matrix addition**

#include <stdio.h>

void main()

{

int arr1[3][3],arr2[3][3],i,j,n;

printf("Please input the numbers");

scanf("%d",&n);

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

{

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

{

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

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

}

}

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

{

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

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

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

} }

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

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

{

printf("\n");

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

printf("%d\t",arr1[i][j]+arr2[3][3]);

}

printf("\n\n");

}

**14. maximum minimum**

#include<stdio.h>

int main()

{

int a[100],i,j,n,max,min;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

max=a[0],min=a[0];

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

if(a[i]>max)

max=a[i];

if(a[i]<min)

min=a[i];

}

printf("Max:%d ",max);

printf("Min:%d ",min);

}

**15. odd even using array**

#include <stdio.h>

void main()

{

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

int i,j=0,k=0,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");

}

**16. reverse order array**

#include<stdio.h>

int main()

{

int a[10],i,j,n;

printf("Please input the numbers");

scanf("%d",&n);

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

printf("Please input the values %d ",i);

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

}

printf("output:\n");

for(i=n-1;i>=0;i--)

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

}

**17. 2nd largest array**

#include <stdio.h>

void main(){

int arr1[50],n,i,j=0,lrg,lrg2nd;

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]);

}

lrg=0;

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

{

if(lrg<arr1[i])

{

lrg=arr1[i];

j = i;

}

}

/\* ignore the largest element and find the 2nd largest element in the array \*/

lrg2nd=0;

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

{

if(i==j)

{

i++; /\* ignoring the largest element \*/

i--;

}

else

{

if(lrg2nd<arr1[i])

{

lrg2nd=arr1[i];

}

} }

printf("The Second largest element in the array is : %d \n\n", lrg2nd);

}

**Code in Function**

**1.** **decimal to binary**

#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;

}

1. **factorial**

#include <stdio.h>

int fact(int);

void main()

{

int sum;

sum=fact(1)/1+fact(2)/2+fact(3)/3+fact(4)/4+fact(5)/5;

printf("\n\n Function : find the sum of 1!/1+2!/2+3!/3+4!/4+5!/5 :\n");

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

}

int fact(int n)

{

int num=0,f=1;

while(num<=n-1)

{

f =f+f\*num;

num++;

}

return f;

}

1. **find out maximum number in array**

#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(" 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;

}

**4. prime or not**

#include<stdio.h>

int PrimeOrNot(int);

int main()

{

int n1,prime;

printf("\n\n Function : check whether a number is prime number or not :\n");

printf(" Input 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<=n1/2)

{

if(n1%i==0)

return 0;

else

i++;

}

return 1;

}

1. **square value**

#include <stdio.h>

double square(double num)

{

return (num \* num);

}

int main()

{

int num;

double n;

printf("\n\n Function : find square of any number :\n");

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

printf("Input any number for square : ");

scanf("%d", &num);

n = square(num);

printf("The square of %d is : %.2f\n", num, n);

return 0;

}

**5. sum value**

#include <stdio.h>

int sum (int, int);//function declaration

int main (void)

{

int total;

printf("\n\n Function : a simple structure of function :\n");

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

total = sum (5, 6);//function call

printf ("The total is : %d\n", total);

return 0;

}

int sum (int a, int b) //function definition

{

int s;

s=a+b;

return s; //function returning a value

}

1. **find out GCD and LCM**

#include <stdio.h>

int LCM (int a,int b);

int GCD (int a,int b);

int main()

{int a,b;

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

printf("LCM is %d\nGCD is %d",LCM (a,b),GCD (a,b));

return 0;}

int LCM (int a,int b)

{ int i;

for(i=2;i<=a\*b;i++) {

if(i%a==0&&i%b==0)

return i;}

}

int GCD (int a,int b)

{ int i;

if(a>b)

{

for(i=b;i>=1;i--)

if(a%i==0&&b%i==0)

return i;

}

else

{

for(i=a;i>=1;i--)

if(a%i==0&&b%i==0)

return i;

}}

**Code in Pointer**

**1. count vowel and consonent**

#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);

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;

}

**2. find maximum in array**

#include <stdio.h>

int main()

{

int arr[100], \*max, n, i;

printf("Enter the number of elements in array\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

max = arr;

\*max = \*arr;

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

{

if (\*(arr+i) > \*max)

{

\*max = \*(arr+i); }

}

printf("Maximum element is %d.\n", \*max);

return 0;

}

**3.find odd or even**

#include <stdio.h>

void main()

{

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

int i,j=0,k=0,n,\*p;

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]);

}

p=arr1;

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

{

if (\*(p+i)%2 == 0)

{

arr2[j] = \*(p+i);

j++;

}

else

{

arr3[k] = \*(p+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");

}

1. **find minimum and average**

#include <stdio.h>

int main()

{

int arr[100], \*min, n, i,\*p,sum=0;

float avg;

printf("Enter the number of elements in array\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

p=arr;

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

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

avg=(float)sum/n; }

min = arr;

\*min = \*arr;

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

{

if (\*(arr+i) < \*min)

{

\*min = \*(arr+i);

} }

printf("Average is %0.2f Minimum element is %d.\n", avg, \*min);

return 0;

}

1. **factorial**

#include <stdio.h>

void findFact(int,int\*);

int main()

{

int fact;

int num1;

printf("\n\n Pointer : Find the factorial of a given number :\n");

printf(" Input a number : ");

scanf("%d",&num1);

findFact(num1,&fact);

printf(" The Factorial of %d is : %d \n\n",num1,fact);

return 0;

}

void findFact(int n,int \*f)

{

int i;

\*f =1;

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

\*f=\*f\*i;

}

1. **largest with recur**

#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("--------------------------------------------------\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;

}

1. **length string**

#include <stdio.h>

int calculateLength(char\*);

void main()

{

char str1[25];

int l;

printf("\n\n Pointer : Calculate the length of the string :\n");

printf(" Input a string : ");

fgets(str1, sizeof str1, stdin);

l = calculateLength(str1);

printf(" The length of the given string %s is : %d ", str1, l-1);

printf("\n\n"); }

int calculateLength(char\* ch) // ch = base address of array str1 ( &str1[0] )

{

int ctr = 0;

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

{

ctr++;

ch++;

}

return ctr;

}

1. **sorting array**

#include <stdio.h>

void main()

{

int \*a,i,j,tmp,n;

printf("\n\n Pointer : Sort an array using pointer :\n");

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

scanf("%d",&n);

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

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

{

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

scanf("%d",a+i);

}

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

{

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

{

if( \*(a+i) > \*(a+j))

{

tmp = \*(a+i);

\*(a+i) = \*(a+j);

\*(a+j) = tmp;

} } }

printf("\n The elements in the array after sorting : \n");

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

{

printf(" element - %d : %d \n",i+1,\*(a+i));

}

printf("\n");

}

1. **sum two num**

#include <stdio.h>

int main()

{

int fno, sno, \*ptr, \*qtr, sum;

printf("\n\n Pointer : Add two numbers :\n");

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

printf(" Input the first number : ");

scanf("%d", &fno);

printf(" Input the second number : ");

scanf("%d", &sno);

ptr = &fno;

qtr = &sno;

sum = \*ptr + \*qtr;

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

return 0;

}

1. **swap fun pointer**

#include <stdio.h>

void swap(int \*n1, int \*n2);

int main()

{

int num1 = 5, num2 = 10;

// address of num1 and num2 is passed

swap( &num1, &num2);

printf("num1 = %d\n", num1);

printf("num2 = %d", num2);

return 0;

}

// pointer n1 and n2 stores the address of num1 and num2 respectively

void swap(int\* n1, int\* n2)

{

int temp;

temp = \*n1;

\*n1 = \*n2;

\*n2 = temp;

}

**Code in Structure**

**1.**

#include<stdio.h>

struct company{

char name[30];

//float term\_marks[3];

float income;

float cost;

float profit;

};

int main()

{

struct company s[3];

int i,j,n;

float profit;

//float sum[3]={0,0,0};

printf("Enter how many companies:");

scanf("%d", &n);

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

{

printf("Name:");

gets(s[j].name);

getchar();

printf("Income:");

scanf("%f",&s[j].income);

printf("Cost:");

scanf("%f",&s[j].cost);

}

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

s[j].profit=s[j].income-s[j].cost;

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

printf("Company %s, Income:%f, Cost %f Profit %f \n",s[j].name,s[j].income,s[j].cost,s[j].profit);

return 0;

}

**2. employee info**

#include<stdio.h>

struct Employee

{

char ename[10];

int sal;

};

struct Employee emp[5];

int i, j;

void ask()

{

for(i = 0; i < 3; i++)

{

printf("\nEnter %dst Employee record:\n", i+1);

printf("\nEmployee name:\t");

scanf("%s", emp[i].ename);

printf("\nEnter Salary:\t");

scanf("%d", &emp[i].sal);

}

printf("\nDisplaying Employee record:\n");

for(i = 0; i < 3; i++)

{

printf("\nEmployee name is %s", emp[i].ename);

printf("\nSlary is %d", emp[i].sal);

}

}

void main()

{

ask();

}

3**. distance detail**

// Program to add two distances which is in feet and inches

#include <stdio.h>

struct Distance

{

int feet;

float inch;

} dist1, dist2, sum;

int main()

{

printf("1st distance\n");

printf("Enter feet: ");

scanf("%d", &dist1.feet);

printf("Enter inch: ");

scanf("%f", &dist1.inch);

printf("2nd distance\n");

printf("Enter feet: ");

scanf("%d", &dist2.feet);

printf("Enter inch: ");

scanf("%f", &dist2.inch);

sum.feet = dist1.feet + dist2.feet;

sum.inch = dist1.inch + dist2.inch;

while (sum.inch >= 12)

{

++sum.feet;

sum.inch = sum.inch - 12;

}

printf("Sum of distances = %d\'-%.1f\"", sum.feet, sum.inch);

return 0;

}

**Code in recursion**

1. factorial

#include<stdio.h>

int findFactorial(int);

int main()

{

int n1,f;

printf("\n\n Recursion : Find the Factorial of a number :\n");

printf(" Input a number : ");

scanf("%d",&n1);

f=findFactorial(n1);//call the function findFactorial for factorial

printf(" The Factorial of %d is : %d\n\n",n1,f);

return 0;

}

int findFactorial(int n)

{

if(n==1)

return 1;

else

return(n\*findFactorial(n-1));// calling the function findFactorial to itself recursively

}

**2. sum range**

#include<stdio.h>

int sumOfRange(int);

int main()

{

int n1;

int sum;

printf("\n\n Recursion : calculate the sum of numbers from 1 to n :\n");

printf(" Input the last number of the range starting from 1 : ");

scanf("%d", &n1);

sum = sumOfRange(n1);

printf("\n The sum of numbers from 1 to %d : %d\n\n", n1, sum);

return (0);

}

int sumOfRange(int n1)

{

int res;

if (n1 == 1)

{

return (1);

} else

{

res = n1 + sumOfRange(n1 - 1); //calling the function sumOfRange itself

}

return (res);

}

**3. fibonacci**

#include<stdio.h>

int term;

int fibonacci(int prNo, int num);

void main()

{

static int prNo = 0, num = 1;

printf("\n\n Recursion : Print Fibonacci Series :\n");

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

printf(" Input number of terms for the Series (< 20) : ");

scanf("%d", &term);

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

printf(" 1 ");

fibonacci(prNo, num);

printf("\n\n");

}

int fibonacci(int prNo, int num)

{

static int i = 1;

int nxtNo;

if (i == term)

return (0);

else

{

nxtNo = prNo + num;

prNo = num;

num = nxtNo;

printf("%d ", nxtNo);

i++;

fibonacci(prNo, num); //recursion, calling the function fibonacci itself

}

return (0);

}