**Programs with recursion**

#include<iostream>

#include<iomanip>

using namespace std;

//fibonacci series

int fibonacci(int n){

    if (n==1 || n==2) return 1;

    if (n>2)

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

}

int fact(int n){

    if (n==1)   return 1;

    else

    return (n\*fact(n-1));

}//factorial program

int gcd(int a, int b){

    if (a%b==0)  return b;

    else return gcd(b,a%b);

}//Greatest commond divisor

void binarysearch(int a[], int low, int high){

    int mid;

    if (low>high)   return -1;

    else{

        mid=(low+high)/2;

        if (a[mid]==x){

            loc=mid;

            return;

        }

        else if(a[mid>x])

        binarysearch(a, low,mid-1,x,loc);

        else

        binarysearch(a,mid+1,high,x,loc);

    }

    return -1;

}

int main(){

    int terms,num, n1,n2, x1;

    int choice;

    char ch='y';

    while (ch=='y'){

        cout<<endl<<"==================="<<endl;

        cout<<"1. FIBONACCI SERIES"<<endl;

        cout<<"2. FACTORIAL NUMBER"<<endl;

        cout<<"3. GCD OF NUMBERS"<<endl;

        cout<<"4. Binary Search"<<endl;

        cout<<"==================="<<endl;

        cout<<endl;

    //take choice from user

    cout<<"Enter your choice : ";

    cin>>choice;

    cout<<endl;

    switch(choice){

        case 1: cout<<"Enter total number of terms for fibocnacci series : ";

                cin>>terms;

                for (int i; i<terms; i++)

                cout<<setw(5)<<fibonacci(i);

                cout<<endl;

                break;

                //fibonacci series

        case 2: cout<<"Enter the number for factorial : ";

                cin>>num;

                cout<<endl<<"Factorial of number is : "<<fact(num)<<endl;

                break;

        case 3: cout<<"Enter two numbers to find GCD : ";

                cin>>n1>>n2;

                cout<<endl<<"GCD is "<<gcd(n1,n2)<<endl;

                break;

        case 4: cout<<"Binary Search of number in array[5]"<<endl;

                int arrayNum[5];

                cout<<endl<<"Enter element of the array : ";

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

                cin>>arrayNum[i];

                cout<<"Enter the element to be searched : ";

                cin>>x1;

                binarysearch(arrayNum, 0, 4, x1, loc);

                if (loc!=-1)

                cout<<"Element found : "<<(loc+1);

                else

                cout<<"Element not found."<<endl;

    }

    cout<<endl<<"\t -> Do you want to continue(y/n) : ";

    cin>>ch;

    cout<<endl;

    }

    return 0;

} //advantages - readibility and shorter code

**MATRIX CLASS AND OBJECTS**

#include <iostream>

#include <iomanip>

using namespace std;

class matrix{

    int a[5][5]; //objects will represent one array only

    int row,col;

    //default private members

    public:

    matrix(int row1=5, int col1=5){

        this.row=row;

        this.col=col;

        for (int i=0; i<row; i++)

        for (int j=0; j<col; j++)

        a[i][j]=0;

    //constructor

    void input();

    void display();

    matrix multiply(matrix o1);

    matrix add(matrix o1);

    matrix subtract(matrix o2);

    matrix transpose();

};

matrix matrix :: transpose(){

    matrix temp;

    temp.row=col;

    temp.col=row;

    for (int i=0; i<temp.row, i++)

    for (int j=0; j<temp.col; j++)

    temp.a[i][j]=a[j][i];

    return temp;

}

void matrix:: display(){

    cout<<"Matrix is"<<endl;

    for (int i=0; i<row; i++){

        for (int j=0; j<col; j++)

        cout<<setw(3)<<a[i][j];

        cout<<endl;

    }

}

void matrix :: input(){

    for(int i=0; i<row;i++){

        cout<<"Enter the elements of row : "<<(i+1);

        for(int j=0;j<col;j++)

        cin>>a[i][j];

    }

}

void matrix::multiply(matrix o1){

    matrix o3;

    if (col==o1.row){

        o3.row=row;

        o3.col=col;

        for (int i=0;i<row;i++){

            for(int j=0; j<o1.col; j++){

                o3.a[i][j]=0;

                for (int k=0; k<col; k++)

                o3.a[i][j]=a[i][j]+a[i][k]\*b[k][j];

            }

        }

        return o3;

    }

    else

    cout<<"Invalid order of matrix multiplication!"<<endl;

}

int main(){

    //A B C

    matrix o1(3,4),o2(4,3), o3(3,3), o4(3,3);     //3 objects

    o1.input();                 //object uses input function

    o2.input();

    o3=o1.multiply(o2);          //object uses multiply function

    o4=o1.transpose();

    cout<<"transpose is "<<endl;

    o4.display();

    return 0;

}