Solution 1:

#include <iostream>

using namespace std;

struct Student{

    int ID,Credit;

    float CGPA;

};

Student userInput()

{

    Student s;

    cout<<"input the ID :";

    cin>>s.ID;

    cout<<"input the credit :";

    cin>>s.Credit;

     cout<<"input the CGPA :";

     cin>>s.CGPA;

     cout<<endl;

     return s;

}

void ShowDetailsID(Student s)

{

    cout<<"The students ID :"<<s.ID<<endl;

}

int main()

{

    Student student[5];

    for (int i = 0; i < 5; i++)

    {

        student[i]= userInput();

    }

    cout<<"Print all the student's IDs whose CGPA is more than 3.60"<<endl;

    for (int i = 0; i < 5; i++)

    {

        if (student[i].CGPA>3.60)

        {

            ShowDetailsID(student[i]);

        }

    }

    cout<<"Print all the student's IDs who have completed more than 60 credits"<<endl;

    for (int i = 0; i < 5; i++)

    {

        if (student[i].Credit>60)

        {

            ShowDetailsID(student[i]);

        }

    }

}

**Solution 2:**

#include <iostream>

using namespace std;

#define MaxSize 8

class MyStack{

    private:

    int top=-1;

    char data[MaxSize];

    public:

    bool isFull()

    {

        if (top>=(MaxSize-1))

        {

            cout<<"Stack is full"<<endl;

            return true;

        }

        else

        {

            return false;

        }

    }

    bool isEmpty()

    {

        if (top==-1)

        {

            cout<<"Stack is Empty"<<endl;

            return true;

        }

        else

        {

            return false;

        }

    }

    void push(string e)

    {

        if(!isFull())

        {

            for (int i = 0; i <e.length(); i++)

            {

                if (top+1<MaxSize){

                {

                    top++;

                    data [top]=e[i];

                }

            }

        }

    }

    }

    void pop()

    {

          if (!isEmpty()){

            char removed\_element = data[top];

            top--;

        }

    }

    void showDetails()

    {

       for (int i = 0; i <= top; i++)

       {

        cout<<data[i];

       }

       cout<<endl;

    }

        void showReverseElement()

        {

        cout << "The reverse elements are: " << endl;

        for (int i = top; i >= 0; i--) {

            cout << data[i];

        }

        cout << endl;

    }

};

int main()

{

    MyStack stk;

    stk.push("Hello");

    cout<<"printing the element"<<endl;

    stk.showDetails();

    stk.showReverseElement();

    return 0;

}

**Solution 3:**

#include <iostream>

using namespace std;

int main()

{

    int n,sum=0;

    cout<<"input size of the array : "<<endl;

    cin>>n;

    int arr[n];

    cout <<"input Numbers"<<endl;

    for(int i=0;i<n;i++)

    {

        cin>>arr[i];

    }

    for(int i = 0; i<n;i++)

    {

        sum =sum + arr[i];

    }

   double average = (double)sum/n;

    cout <<"Average :"<<average;

}

**Solution 4:**

#include <iostream>

using namespace std;

int main() {

    int mat1[2][2], mat2[2][2], mat3[2][2], mat4[2][2], sum[2][2];

    cout << "Enter Elements of First Matrix: ";

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++)

            cin >> mat1[i][j];

    }

    cout << "Enter Elements of Second Matrix: ";

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++)

            cin >> mat2[i][j];

    }

    cout << "Enter Elements of Third Matrix: ";

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++)

            cin >> mat3[i][j];

    }

    cout << "Enter Elements of Fourth Matrix: ";

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++)

            cin >> mat4[i][j];

    }

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++)

            sum[i][j] = mat1[i][j] + mat2[i][j] + mat3[i][j] + mat4[i][j];

    }

    cout << "The sum of the four matrices is:\n";

    for (int i = 0; i < 2; i++) {

        for (int j = 0; j < 2; j++) {

            cout << sum[i][j] << " ";

        }

        cout << endl;

    }

    return 0;

}