NAME : Ridhima Goel

SUBGROUP : 2C32

ROLL NO : 1024030466

DSA ASSIGNMENT

QUES 1) Develop a Menu driven program to demonstrate the following operations of Arrays

——MENU——-

1.CREATE

2.DISPLAY

3.INSERT

4.DELETE

5.LINEAR SEARCH

6.EXIT

Solution :

#include<iostream>

using namespace std;

int arr[1000];

int n;

void display(){

    for(int i = 0;i<n;i++){

        cout<<arr[i];

    }

}

main(){

    int option;

    int choice;

    cout<<"enter the size of array: ";

    cin>>n;

    for(int i = 0;i<n;i++){

        cout<<"Element "<<i+1<<":";

        cin>>arr[i];

    }

    display();

    do{

    cout<<"\nOption 1 : Insert element in array";

    cout<<"\nOption 2 : Delete an element in array";

    cout<<"\nOption 3 : Linear search ";

    cout<<"\nOption 4 : Exit";

    cout<<"\nWhich option you would choose ";

    cin>>option;

    switch(option){

        case 1 :

        int element,pos;

        cout<<"enter the element you want to insert :";

        cin>>element;

        cout<<"enter the pos where you want to insert element :";

        cin>>pos;

        for(int i = n-1;i>=pos-1;i--){

            arr[i+1] = arr[i];

        }

        arr[pos-1]=element;

        n++;

        display();

        break;

        case 2 :

        int index;

        display();

        cout<<"\nenter the index of the element where you want to wish to delete : ";

        cin>>index;

        if(index > n +1 ){

            cout<<"deletion not possible";

        }

        else{

            for(int i = index-1;i<n-1;i++){

            arr[i]=arr[i+1];

            }

        }

        n--;

        display();

        break;

        case 3 :

        int item;

        cout<<"Enter the element you want to search ";

        cin>>item;

        for(int i = 0;i<n;i++){

            if(arr[i]==item){

                cout<<"element found at position :"<<i+1;

            }

            else{

                cout<<"element didnt found";

            }

        }

    }

    cout<<"\npress 0 for exit..press 1 for further ";

    cin>>choice;

}while(choice == 1);

}

Output :

A screen shot of a computer

AI-generated content may be incorrect.



QUES 2) Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

SOLUTION :

#include<iostream>

using namespace std;

int main(){

    int n;

    int pos;

    int arr[1000];

    cout<<"Enter the size of array : ";

    cin>>n;

    for(int i = 0;i < n;i++){

        cout<<"Element "<<i+1<<":";6

        cin>>arr[i];

    }

    //sorting

    for(int i = 0;i<n-1;i++){

        for(int j = 0;j<n-i-1;j++){

            if(arr[j]>arr[j+1]){

               int  temp=arr[j];

               arr[j]=arr[j+1];

               arr[j+1]=temp;

            }

        }

    }

    for(int i = 1;i<n;){

        if(arr[i-1]==arr[i]){

           for(int j = i; j<n-1;j++){

                arr[j]=arr[j+1];

            } n--;

        }

        else{

            i++;

        }

    }

    for(int i = 0;i<n;i++){

        cout<<arr[i];

    }

}

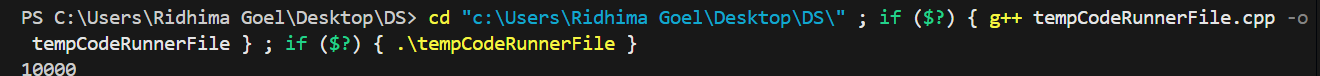
OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.

QUES 3)Predict the Output of the following program int main() {

int i; int arr[5] = {1}; for (i = 0; i < 5; i++) printf("%d",arr[i]); return 0; }



QUES 4) Implement the logic to a. Reverse the elements of an array b. Find the matrix multiplication c. Find the Transpose of a Matrix

#include<iostream>

using namespace std;

main(){

    int option;

    cout<<"Option 1 : Reverse the array ";

    cout<<"\nOption 2 : Matrix multiplication";

    cout<<"\nOption 3 : Transpose of matrix ";

    cout<<"\nEnter the option";

    cin>>option;

    switch(option){

        case 1 :{

        int arr[1000];

        int n;

        cout<<"Enter the size of array";

        cin>>n;

        for(int i = 0;i<n;i++){

            cout<<"element "<<i+1<<":";

            cin>>arr[i];

        }

        int start = 0;

        int end = n-1;

        while(start < end){

            int temp = arr[start];

            arr[start] = arr[end];

            arr[end] = temp;

            start++;

            end--;

        }

        for(int i = 0;i<n;i++){

            cout<<arr[i];

        }

        break;}

        case 2:{

        int r,c;

        int r1,c1;

        int arr1[100][100],arr2[100][100];

        int arr3[100][100];

        cout<<"Enter no of rows in matrix 1: ";

        cin>>r;

        cout<<"Enter no of columns in matrix 1: ";

        cin>>c;

        cout<<"Enter elements for matrix 1 \n";

        for(int i = 0;i<r;i++){

            for(int j = 0;j<c;j++){

                cout<<"Enter element at "<<"["<<i<<"]"<<"["<<j<<"]";

                cin>>arr1[i][j];

            }

        }

        cout<<"Enter no of rows in matrix 2 : ";

        cin>>r1;

        cout<<"Enter no of columns in matrix 2 : ";

        cin>>c1;

        cout<<"Enter elements for matrix 2 \n";

        for(int i = 0;i<r1;i++){

            for(int j = 0;j<c1;j++){

                cout<<"Enter element at "<<"["<<i<<"]"<<"["<<j<<"]";

                cin>>arr2[i][j];

            }

        }

        if(c!=r1){

            cout<<"Multiplication cant be done as no of columns of first matrix is not same as no of rows in second matrix";

        }

        for(int i = 0;i<r;i++){

            for(int j = 0;j<c1;j++){

                arr3[i][j]=0;

            }

        }

        for(int i = 0;i<r;i++){

            for(int j = 0;j<c1;j++){

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

                    arr3[i][j]+=arr1[i][k]\*arr2[k][j];

                }

            }

        }

        cout<<"resultant matrix";

         for(int i = 0;i<r;i++){

            for(int j = 0;j<c1;j++){

                cout<<arr3[i][j];

                cout<<" ";

            }cout<<endl;

        }

        break;}

        case 3 :

        {

        int arr4[100][100];

        int r3,c3;

        cout<<"Enter the no of rows : ";

        cin>>r3;

        cout<<"Enter the no of columns : ";

        cin>>c3;

        cout<<"Enter elements for matrix \n";

        for(int i = 0;i<r3;i++){

            for(int j = 0;j<c3;j++){

                cout<<"Enter element at "<<"["<<i<<"]"<<"["<<j<<"]";

                cin>>arr4[i][j];

            }

        }

        for(int i = 0;i<c3;i++){

            for(int j = 0;j<r3;j++){

                cout<<arr4[j][i];

                cout<<" ";

            }

            cout<<endl;

        }

    }}

}

Output:

A screen shot of a computer

AI-generated content may be incorrect.



A screen shot of a computer

AI-generated content may be incorrect.

5) Write a program to find sum of every row and every column in a two-dimensional array.

#include<iostream>

using namespace std;

main(){

    int arr[100][100];

    int r,c;

    cout<<"Enter the no of  rows";

    cin>>r;

    cout<<"Enter the no of columns";

    cin>>c;

    for(int i = 0;i<r;i++){

        for(int j =0;j<c;j++){

             cout<<"Enter element at "<<"["<<i<<"]"<<"["<<j<<"] :";

            cin>>arr[i][j];

        }

    }

     for(int i = 0;i<r;i++){

        for(int j =0;j<c;j++){

          cout<<arr[i][j];

          cout<<" ";

        }cout<<endl;

    }

    int option;

    cout<<"Option 1 : Row-wise sum";

    cout<<"\nOption 2 : column wise sum";

    cout<<"\nOption 3 : Sum of whole matrix";

    cout<<"Enter the option: ";

    cin>>option;

    switch(option){

        case 1:

    int sum ;

    for(int i = 0;i<r;i++){

       sum = 0;

        for(int j =0;j<c;j++){

        sum+=arr[i][j];

        }

        cout<<"Sum of "<<i+1<<"row : "<<sum;

        cout<<endl;

    }

    break;

    case 2 :

      int sum2;

    for(int j = 0;j<c;j++){

       sum2 = 0;

        for(int i =0;i<r;i++){

        sum2+=arr[i][j];

        }

        cout<<"Sum of "<<j+1<<"column : "<<sum2;

        cout<<endl;}

        break;

    case 3:

    int sum1 = 0;

    for(int i =0;i<r;i++){

        for(int j = 0;j<c;j++){

            sum1+=arr[i][j];

        }

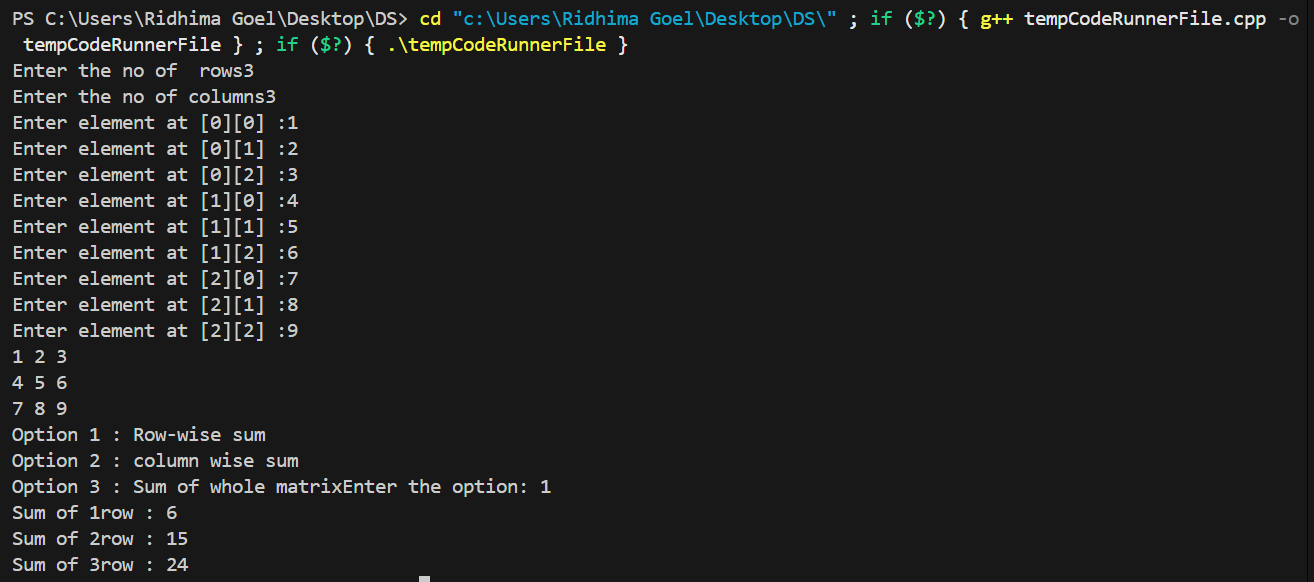
    }

    cout<<"Sum of whole matrix : "<<sum1;

    }

}

Output:



A screenshot of a computer

AI-generated content may be incorrect.

