ASSIGNMENT 1

ADHYA RAZDAN

1024170337

**Q1.**

#include <iostream>

using namespace std;

const int MAX=100;

int arr[MAX];

int n=0;

void CREATE(){

    int arr[MAX];

    cout<<"Enter number of elements ";

    cin>>n;

    cout<<"Enter elements\n";

    for(int i=0;i<n;i++){

        cin>>arr[i];

    }

}

void DISPLAY(){

    int arr[MAX];

    if (n==0){

        cout<<"Empty array";

    }

    cout<<"Array elements are ";

    for (int i=0;i<n;i++){

        cout<<arr[i]<<" ";

    }

    cout<<endl;

}

void INSERT(){

    int arr[MAX];

    if (n>=MAX){

        cout<<"Array is full, cannot insert more elements";

        return;

    }

    int val,pos;

    cout<<"Enter position for new element ";

    cin>>pos;

    if (pos<0||pos>n){

        cout<<"invalid";

        return;

    }

    cout<<"Enter new element ";

    cin>>val;

    for (int i=n;i>pos;i--){

        arr[i]=arr[i-1];

    }

    arr[pos]=val;

    n++;

    cout<<"element inserted successfully";

}

void DELETE(){

    int arr[MAX];

    if (n==0){

        cout<<"Empty array";

        return;

    }

    int pos;

    cout<<"Enter position to delete";

    cin>>pos;

    if (pos<0||pos>=n){

        cout<<"invalid";

        return;

    }

    for(int i=pos;i<n-1;i++){

        arr[i]=arr[i+1];

    }

    n++;

    cout<<"Deleted successfully\n";

}

void LINEAR\_SEARCH(){

    int arr[MAX];

    if(n==0){

        cout<<"Empty array\n";

        return;

    }

    int key,found=0;

    cout<<"Enter value to find";

    cin>>key;

    for (int i=0;i<n;i++){

        if (arr[i]==key){

            cout<<"Key found at "<<i<<endl;

            found=1;

            break;

        }

    }

    if (!found){

        cout<<"element not found";

        return;

    }

}

int main(){

    int choice;

    do{

        cout<<"\n-----MENU-----\n";

        cout<<"1.CREATE\n2.DISPLAY\n3.INSERT\n4.DELETE\n5.LINEAR SEARCH\n6.EXIT\n";

        cout<<"Enter your choice";

        cin>>choice;

        switch(choice){

            case 1:

            CREATE();

            break;

            case 2:

            DISPLAY();

            break;

            case 3:

            INSERT();

            break;

            case 4:

            DELETE();

            break;

            case 5:

            LINEAR\_SEARCH();

            break;

            case 6:

            cout<<"Exiting...\n";

            break;

            default:

            cout<<"INVALID!!!";

        }

    }while(choice!=6);

    return 0;

}

**Q2.**

#include <iostream>

using namespace std;

// Bubble Sort function

void bubbleSort(int arr[], int n) {

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

bool swapped = false; // Optimization: check if already sorted

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

if (arr[j] > arr[j + 1]) {

// Swap

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

swapped = true;

}

}

if (!swapped) break; // Stop if no swap happened

}

}

int delduplicate(int arr[],int n){

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

return n;

}

int j=0;

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

if (arr[j]!=arr[i]){

j++;

arr[j]=arr[i];

}

}

return j+1;

}

int main(){

int arr[]={1,22,34,4,5,16,27,8,19,10};

int n=sizeof(arr)/sizeof(arr[0]);

n=delduplicate(arr,n);

cout<<"array is ";

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

cout<<arr[i]<<" ";

}

cput<<endl;

return 0;

}

**Q4. a.**

#include <iostream>

using namespace std;

void reverseArray(int arr[], int n) {

int start = 0, end = n - 1;

while (start < end) {

// Swap elements

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

void displayArray(int arr[], int n) {

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

cout << arr[i] << " ";

}

cout << endl;

}

int main() {

int n;

cout << "Enter number of elements: ";

cin >> n;

int arr[100]; // assuming max size 100

cout << "Enter " << n << " elements:\n";

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

cin >> arr[i];

}

cout << "Original array: ";

displayArray(arr, n);

reverseArray(arr, n);

cout << "Reversed array: ";

displayArray(arr, n);

return 0;

}

**b. and c.**

#include <iostream>

using namespace std;

void multiplyMatrices(int a[10][10], int b[10][10], int result[10][10], int r1, int c1, int r2, int c2) {

if (c1 != r2) {

cout << "Matrix multiplication not possible. (c1 != r2)\n";

return;

}

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

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

result[i][j] = 0;

}

}

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

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

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

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

}

}

}

}

void transposeMatrix(int a[10][10], int result[10][10], int r, int c) {

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

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

result[j][i] = a[i][j];

}

}

}

void displayMatrix(int a[10][10], int r, int c) {

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

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

cout << a[i][j] << " ";

}

cout << endl;

}

}

int main() {

int r1, c1, r2, c2;

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

cout << "Enter rows and columns of first matrix: ";

cin >> r1 >> c1;

cout << "Enter elements of first matrix:\n";

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

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

cin >> a[i][j];

}

}

cout << "Enter rows and columns of second matrix: ";

cin >> r2 >> c2;

cout << "Enter elements of second matrix:\n";

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

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

cin >> b[i][j];

}

}

cout << "\nMatrix Multiplication:\n";

multiplyMatrices(a, b, result, r1, c1, r2, c2);

if (c1 == r2) displayMatrix(result, r1, c2);

cout << "\nTranspose of first matrix:\n";

transposeMatrix(a, result, r1, c1);

displayMatrix(result, c1, r1);

cout << "\nTranspose of second matrix:\n";

transposeMatrix(b, result, r2, c2);

displayMatrix(result, c2, r2);

return 0;

}

**Q5.**

#include <iostream>

using namespace std;

int main() {

int rows, cols;

int arr[10][10];

cout << "Enter number of rows: ";

cin >> rows;

cout << "Enter number of columns: ";

cin >> cols;

cout << "Enter elements of the matrix:\n";

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> arr[i][j];

}

}

// Sum of each row

cout << "\nSum of each row:\n";

for (int i = 0; i < rows; i++) {

int rowSum = 0;

for (int j = 0; j < cols; j++) {

rowSum += arr[i][j];

}

cout << "Row " << i + 1 << " sum = " << rowSum << endl;

}

// Sum of each column

cout << "\nSum of each column:\n";

for (int j = 0; j < cols; j++) {

int colSum = 0;

for (int i = 0; i < rows; i++) {

colSum += arr[i][j];

}

cout << "Column " << j + 1 << " sum = " << colSum << endl;

}

return 0;

}