Q1)

#include <iostream>

using namespace std;

int main() {

int arr[100];

int n = 0;

int num;

while (true) {

cout << "\nEnter a number (1-6):" << endl;

cout << "1. Create array" << endl;

cout << "2. Display array" << endl;

cout << "3. Insert element" << endl;

cout << "4. Delete element" << endl;

cout << "5. Linear search" << endl;

cout << "6. Exit" << endl;

cout << "Enter your choice: ";

cin >> num;

switch (num) {

case 1:

{

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

cin >> n;

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

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

cin >> arr[i];

}

break;

}

case 2:

{

if (n == 0) {

cout << "Array not created yet." << endl;

} else {

cout << "Array elements are: ";

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

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

}

cout << endl;

}

break;

}

case 3:

{

if (n == 0) {

cout << "Array not created yet. Please create it first." << endl;

break;

}

int pos, element;

cout << "Enter the index to insert (0 to " << n << "): ";

cin >> pos;

if (pos < 0 || pos > n) {

cout << "Invalid index!" << endl;

break;

}

cout << "Enter the element to insert: ";

cin >> element;

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

arr[i] = arr[i - 1];

}

arr[pos] = element;

n++;

cout << "Updated array is: ";

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

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

}

cout << endl;

break;

}

case 4:

{

if (n == 0) {

cout << "Array not created yet." << endl;

break;

}

int pos;

cout << "Enter the index to remove (0 to " << n - 1 << "): ";

cin >> pos;

if (pos < 0 || pos >= n) {

cout << "Invalid index!" << endl;

break;

}

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

arr[i] = arr[i + 1];

}

n--;

cout << "Updated array is: ";

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

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

}

cout << endl;

break;

}

case 5: {

if (n == 0) {

cout << "Array not created yet." << endl;

break;

}

int element;

cout << "Enter the element to search: ";

cin >> element;

int pos1 = -1;

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

if (arr[i] == element) {

pos1 = i;

break;

}

}

if (pos1 == -1) {

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

} else{

cout << "Element found at index: " << pos1 << endl;

}

break;

}

case 6: {

cout << "Exiting program." << endl;

return 0;

}

default:

cout << "Invalid choice! Please enter between 1 and 6." << endl;

}

}

return 0;

}

Q2)

#include<iostream>

using namespace std;

int main(){

int a[]={2,3,5,2,4};

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

bool visited[n]={false};

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

bool duplicate=false;

if(visited[i]==true)

continue;

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

if(a[i]==a[j]){

visited[j]=true;

duplicate=true;

}

}

if(duplicate)

cout<<a[i]<<" is duplicate in the array\n";

}

cout << "Unique elements:\n";

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

if (!visited[i]) {

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

}

}

return 0;

}

Q3) 10000

Q4) A

#include <iostream>

using namespace std;

int main() {

int arr[5] = {10, 20, 30, 40, 50};

int n = 5;

for (int i = 0; i < n / 2; i++) {

int temp = arr[i];

arr[i] = arr[n - 1 - i];

arr[n - 1 - i] = temp;

}

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

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

return 0;

}

B)

#include <iostream>

using namespace std;

int main() {

int a[2][2] = {{1, 2}, {3, 4}};

int b[2][2] = {{5, 6}, {7, 8}};

int c[2][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

c[i][j] = 0;

for (int k = 0; k < 2; k++) {

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

}

}

}

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++)

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

cout << endl;

}

return 0;

}

C)

#include <iostream>

using namespace std;

int main() {

int a[2][3] = {{1, 2, 3}, {4, 5, 6}};

int transpose[3][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

transpose[j][i] = a[i][j];

}

}

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

for (int j = 0; j < 2; j++)

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

cout << endl;

}

return 0;

}

Q5)

#include <iostream>

using namespace std;

int main() {

int rows, cols;

int arr[10][10];

cin >> rows >> cols;

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

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

cin >> arr[i][j];

}

}

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;

}

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;

}