Q.1:

#include <iostream>

using namespace std;

const int Size = 100;

int arr\_size;

int Trace(int data[][Size])

{

int sum = 0;

for (int i = 0; i < arr\_size; i++) { // Output of matrix

sum += data[i][i];

}

return sum;

}

int main()

{

cout << "Enter size of row/column of square matrix: ";

cin >> arr\_size;

int arr[Size][Size];

for (int i = 0; i < arr\_size; i++) { // Getting user input for array

for (int j = 0; j < arr\_size; j++) {

cout << "Enter [" << i << "] [" << j << "] value: ";

cin >> arr[i][j];

}

}

cout << "\nThe given matrix is:\n";

for (int i = 0; i < arr\_size; i++) { // Output of matrix

for (int j = 0; j < arr\_size; j++) {

cout << arr[i][j] << ' ';

}

cout << '\n';

}

cout << "\nSum of diagonals is: " << Trace(arr) << endl;

system("pause");

return 0;

}



Q.2:

#include <iostream>

using namespace std;

const int N = 5;

void Upper\_half(int A[N][N])

{

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

for (int j = 0; j < i; j++) { // spaces before print

cout << ' ';

}

for (int j = i; j < N; j++) { // print row

cout << A[i][j];

}

cout << '\n';

}

}

int main()

{

int arr[N][N] = { {2,3,1,5,0},

{7,1,5,3,1},

{2,5,7,8,1},

{0,1,5,0,1},

{3,4,9,1,5} };

Upper\_half(arr);

system("pause");

return 0;

}



Q.3:

#include <iostream>

using namespace std;

bool Exists(int data[][6],int pattern[][3])

{

bool found;

for (int i = 0; i < 6 - 3; i++) { // 6 - 3 indicates that after this value, we cannot find array since search size(3x3) exceeds data limits.

for (int j = 0; j < 6 - 3; j++) { // (sizeof data - sizeof pattern)

if (data[i][j] == pattern[0][0]) {

found = true;

for (int x = 0; x < 3; x++) { // initiating search using x and y as index values

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

if (data[i + x][j + y] != pattern[x][y]) {

found = false;

break; // if any wrong value, quit search of that column

}

}

if (!found) // if any wrong value, quit search of that area

break;

}

if (found) // quit function if found pattern within data

return true;

}

}

}

return false; // checked all possible combinations and not found

}

int main()

{

int data[6][6] = { {1,2,7,8,9,6},

{2,2,3,4,5,6},

{3,2,3,4,5,6},

{4,2,3,4,5,6},

{5,2,9,8,7,6},

{6,2,7,4,5,6} };

int pattern[3][3] = { {3,4,5},

{3,4,5},

{3,4,5}};

cout << boolalpha << Exists(data, pattern) << endl;

system("pause");

return 0;

}

