Omer Ahmer

Assignment 3

Part 1:

/\*

Description: This program gets temperatures from user, sorts the array filled with the temperatures, and returns the average

Author: Omer Ahmer

COMSC 165-5065

Date: 9/10/22

Status: Complete

\*/

#include <iostream>

#include <iomanip>

using namespace std;

void sorting(int size, int array[size]);

void displayAvg(int size, int array[size]);

int main() {

int count;

const int MAX = 10;

cout << "Enter number of temperatures: " << endl;

cin >> count;

if (count > MAX) {

cout << "You have entered too many temperatures. Please try again:" << endl;

cin >> count;

}

int temp[count];

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

cout << "Enter temperature " << i+1 << ":" << endl;

cin >> temp[i];

}

sorting(count, temp);

displayAvg(count, temp);

}

void sorting(int size, int array[size]) {

int temp, i, j;

for (i = 0; i < size; i++) {

for (j = 0; j < size - 1; j++) {

if (array[j] > array[i]) {

temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

}

cout << "The temperatures you have entered in ascending order are: ";

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

cout << array[k] << endl;

}

}

void displayAvg(int size, int array[size]) {

double sum = 0, avg;

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

sum += array[i];

}

avg = sum/size;

cout << "The average of the temperatures you entered is: " << setprecision(4) << avg << endl;

}

Text

Description automatically generated

Part 2:

/\*

Description: This program runs a simulation of a 2-player tic-tac-toe game

Author: Omer Ahmer

COMSC 165-5065

Date: 9/10/22

Status: Complete

\*/

#include <iostream>

using namespace std;

// Determines whose turn it is

void player\_turn();

// Displays board

void display();

// Initialize empty board with asterisks

char board[3][3] = {{'\*','\*','\*'}, {'\*','\*','\*'}, {'\*','\*','\*'}};

bool draw = false;

int main() {

bool gameover;

// Checks for empty boxes

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

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

if (board[i][j] != 'X' || board[i][j] != 'O') {

gameover = false;

}

}

}

// Checks rows and columns

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

if ((board[i][0] == board[i][1] == board[i][2]) || (board[0][i] == board[1][i] == board[2][i])) {

gameover = true;

}

}

if (board[0][0] == board [1][1] == board[2][2] || board[0][2] == board[1][1] && board[0][2] == board[2][0]) {

gameover = true;

}

while (gameover) {

display();

player\_turn();

gameover = false;

}

if (draw == true) {

cout << "Draw!" << endl;

}

}

void player\_turn() {

char turn = 'X';

int input\_row, input\_column;

if (turn == 'X') {

cout << "Player 1:" << endl;

cout << "Enter row and column to place an X:" << endl;

cout << "Row: ";

cin >> input\_row;

cout << "Column: ";

cin >> input\_column;

if ((turn == 'X') && (board[input\_row-1][input\_column-1] != 'X') && (board[input\_row-1][input\_column-1] != 'O')) {

board[input\_row][input\_column] = 'X';

turn = 'O';

display();

}

} else if (turn == 'O') {

cout << "Player 2:" << endl;

cout << "Enter row and column to place an O:" << endl;

cout << "Row: ";

cin >> input\_row;

cout << "Column: ";

cin >> input\_column;

if ((turn == 'X') && (board[input\_row-1][input\_column-1] != 'X') && (board[input\_row-1][input\_column-1] != 'O')) {

board[input\_row][input\_column] = 'O';

turn = 'X';

display();

}

}

}

void display() {

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

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

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

}

cout << endl << endl;

}

}

Part 3:

/\*

Description: This program converts the existing arrays to vectors

Author: Omer Ahmer

COMSC 165-5065

Date: 9/10/22

Status: Complete

\*/

#include <iostream>

#include <vector>

using namespace std;

bool testPIN(const vector<int>, const vector<int>);

int main()

{

vector<int> pin\_1 {2, 4, 1, 8, 7, 9, 0}; // Base set of values.

vector<int> pin\_2 {2, 4, 6, 8, 7, 9, 0}; // Only 1 element is different from pin1.

vector<int> pin\_3 {1, 2, 3, 4, 5, 6, 7}; // All elements are different from pin1.

if (testPIN(pin\_1, pin\_2))

cout << "Error: pin\_1 and pin\_2 report to be the same.\n";

else

cout << "Success: pin\_1 and pin 2 are different.\n";

if (testPIN(pin\_1, pin\_3))

cout << "ERROR: pin\_1 and pin\_3 report to be the same.\n";

else

cout << "SUCCESS: pin\_1 and pin\_3 are different.\n";

if (testPIN(pin\_1, pin\_2))

cout << "SUCCESS: pin\_1 and pin\_1 report to be the same.\n";

else

cout << "ERROR: pin\_1 and pin\_1 report to be different.\n";

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// The following function accepts two int arrays. The arrays are \*

// compared. If they contain the same values, true is returned. \*

// If the contain different values, false is returned.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bool testPIN(const vector<int> customerPIN, const vector<int> databasePIN)

{

for(int index = 0; index < customerPIN.size(); index++)

{

if (customerPIN[index] != databasePIN[index])

return false; // We've found two different values.

}

return true; // If we make it this far, the values are the same.

}

Text

Description automatically generated