#include <iostream>

using namespace std;

bool isDistinct(int\* array, int num);

int main()

{

    int \*distinctNumbers = new int[10]; // Array of length 10;

    int number;                 // User input

    int count = 0;          // Number of distinct numbers

        // Prompt the user to enter ten numbers

        cout << "Enter ten numbers: ";

        for (int i = 0; i < 10; i++)

        {

            cin >> number;

            // Test if num is distinct

            if (isDistinct(distinctNumbers, number))

            {

                distinctNumbers[i] = number;

                // Increment count

            }

        }

    cout << "The distinct number array is: ";

    for (int i = 0; i < 10; i++)

    {

        cout << distinctNumbers[i] << ", ";

    }

}

    /\*\* Method isDistinct returns true if number is not in array false otherwise \*/

bool isDistinct(int\* array, int num)

{

    for (int i = 0; i < 10; i++)

    {

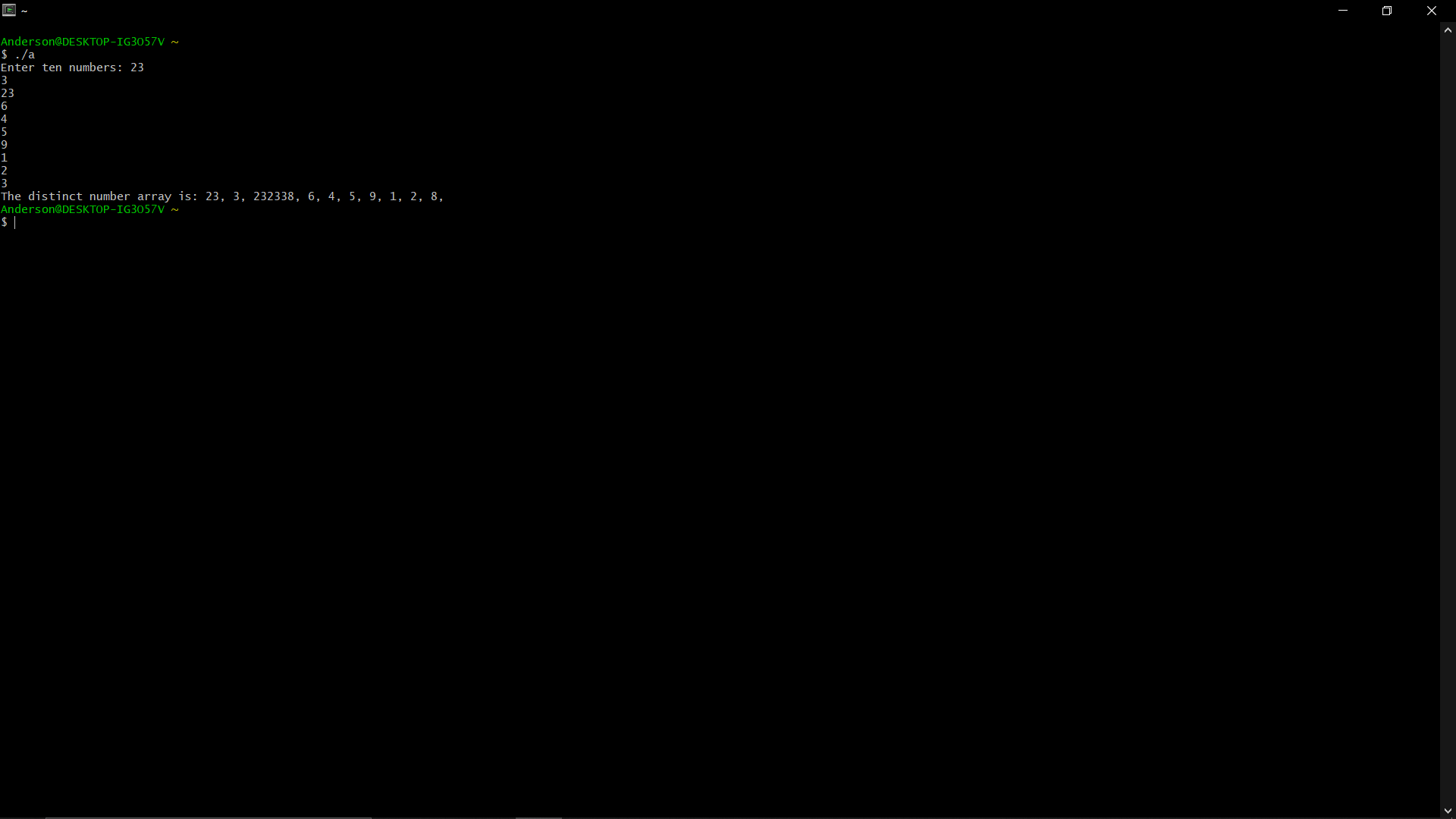
        if (num == array[i])

        return false;

    }

    return true;

}



#include <iostream>

#include <ctime>

using namespace std;

int main()

{

    //Declare variables

    const int LENGTH = 10;

    int \*count = new int[LENGTH];

    int j = 0;

    //Seeding rand for better performance

    srand(time(0));

    //Initialize the array to zero(possible to never generate one or more values from 0-9 so zero will output)

    for (int i = 0; i < LENGTH; i++)

    {

        count[i] = 0;

    }

    //Generate random variables and tally each in the array

    for (int i = 0; i < 100; i++)

    {

        j = rand() % 10;

        count[j]++;

    }

    //Output result

    for (int i = 0; i < LENGTH; i++)

    {

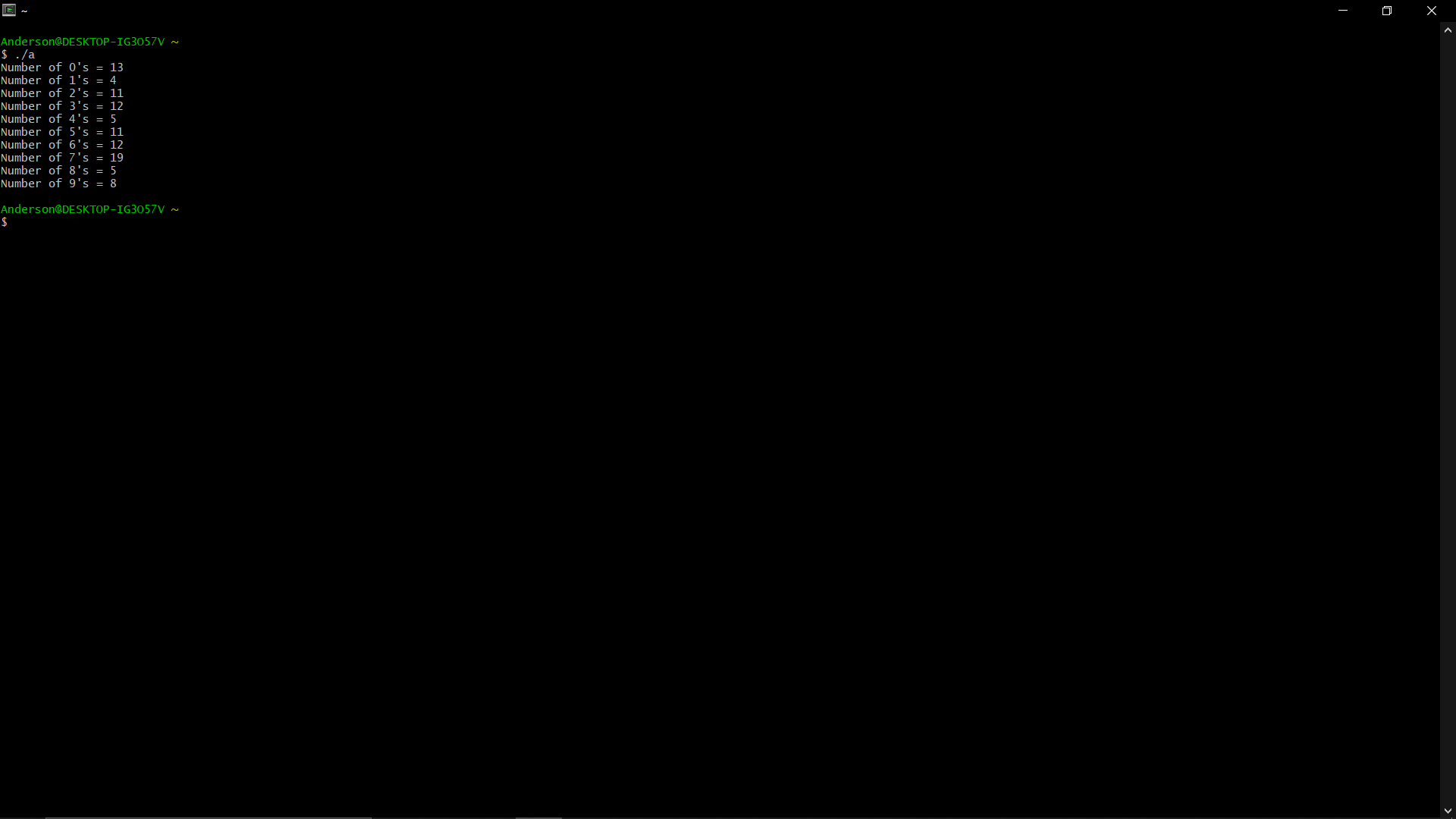
        cout << "Number of " << i << "'s = " << count[i] << endl;

    }

    delete[] count;

    return 0;

}



#include <iostream>

#include <iomanip>

/\*(Sort students) Write a program that prompts the user to enter the number of students, the students’ names, and their scores, and

 \*prints student names in decreasing order of their scores.

 \*/

using namespace std;

// Function prototypes

void sort(double\*, int, string\*);

int main()

{

    int students;

    int numTestScores;  // The number of test scores

    double \*testScoresPtr;  // To point to an array

    // Get the number of test scores.

    cout << "\nHow many students and scores will you enter? \n";

    cin >> students;

    // Validate the input.

    while (students < 0)

    {

        cout << "The number cannot be negative.\n";

        cout << "Enter another number: ";

        cin >> students;

    }

    // Allocate an array to hold the test scores.

    testScoresPtr = new double[students];

    string \*studentNames= new string[students];

    // Fill the arrays with test scores and names.

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

    {

        cout << "\nEnter the name for student " << i + 1 << ": ";

        cin >> studentNames[i];

        cout << "\nEnter test score " << (i + 1) << ": ";

        cin >> testScoresPtr[i];

        cout << endl;

        // Validate the input.

        while (testScoresPtr[i] < 0)

        {

            cout << "Negative scores are not allowed.\n";

            cout << "Enter another score for this test: ";

            cin >> testScoresPtr[i];

        }

    }

    // Sort the test scores.

    sort(testScoresPtr, students, studentNames);

    // Display the resulting data.

    cout << "\nThe students names and scores in decreasing order are:\n\n";

    for (int j = students - 1; j >= 0; j--)

    {

        cout << studentNames[j] << ": ";

        cout << fixed << setprecision(2) << testScoresPtr[j] << endl;

    }

    // Free the dynamically-allocated memory.

    delete[] testScoresPtr;

    delete[] studentNames;

    testScoresPtr = 0;

    return 0;

}

void sort(double\* score, int size, string\* names)

{

    bool swap;

    int temp;

    string nameTemp;

    do

    {

        swap = false;

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

        {

            if (score[i] > score[i + 1])

            {

                temp = score[i];

                score[i] = score[i + 1];

                score[i + 1] = temp;

                nameTemp = names[i];

                names[i] = names[i+1];

                names[i+1] = nameTemp;

                swap = true;

            }

        }

    } while (swap);

}



#include <iostream>

/\*The classic Eight Queens puzzle is to place eight queens on a chessboard such that no two queens can attack each other

 \*(i.e., no two queens are on the same row, same column, or same diagonal). There are many possible solutions. Write a program that displays one such solution.

 \*A  sample output is shown below:

 \*/

using namespace std;

int main()

{

    char queenArray[8][8] = {

                                {81, 32, 32, 32, 32, 32, 32, 32},

                                {32, 32, 32, 32, 81, 32, 32, 32},

                                {32, 32, 32, 32, 32, 32, 32, 81},

                                {32, 32, 32, 32, 32, 81, 32, 32},

                                {32, 32, 81, 32, 32, 32, 32, 32},

                                {32, 32, 32, 32, 32, 32, 81, 32},

                                {32, 81, 32, 32, 32, 32, 32, 32},

                                {32, 32, 32, 81, 32, 32, 32, 32},

                            };

    for (int i = 0; i < 8; i++)

    {

        for (int j = 0; j < 8; j++)

        {

            cout << "|" << queenArray[i][j];

        }

        cout << "|" << endl;

    }

}

