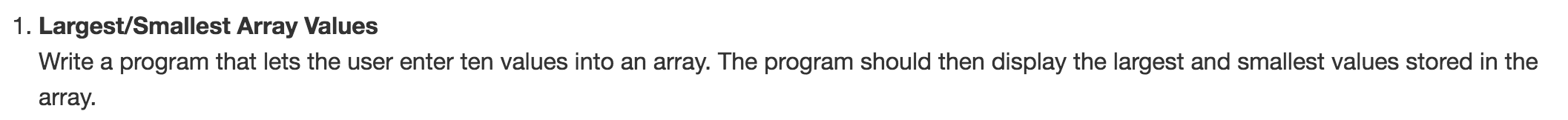
COMSC-110 Lab-6 Due: July-12-2020

Format:

1- After each problem statement, copy and paste the source code.

2- After the source code, paste the screen shot of the result.

3- Submit either word or pdf file (ONLY ONE FILE – NO MAC PAGE FILES)



Answer:

#include <iostream>

using namespace std;

int main() {

const int SIZE = 10;

int arr[5] = {0}; // initializing the array to zeros;

for (auto &temp : arr){

cout << "Enter a number " << endl;

cin >> temp;

}

int min = arr[0];

int max = arr[0];

for (int element : arr){

if (element > max) max = element;

if (element < min) min = element;

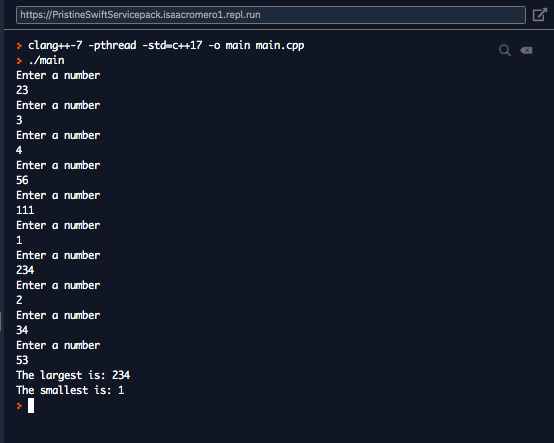
}

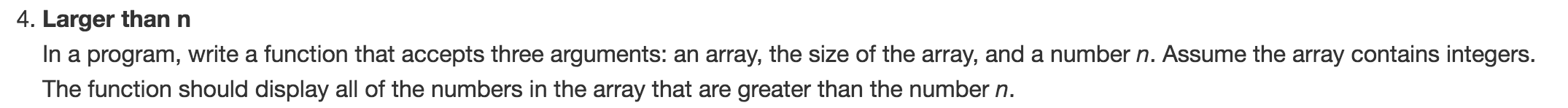
cout << "The largest is: " << max << endl;

cout << "The smallest is: " << min << endl;

return 0;

}





Answer:

#include <iostream>

using namespace std;

void greaterThan(int arr[], int SIZE, int n);

int main() {

const int SIZE = 5;

int arr[SIZE] = {12,23,12,45,55};

greaterThan(arr, SIZE, 20);

}

void greaterThan(int arr[], int SIZE, int n){

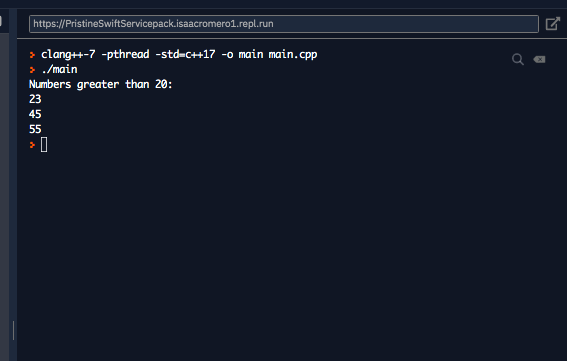
cout << "Numbers greater than " << n << ":" << endl;

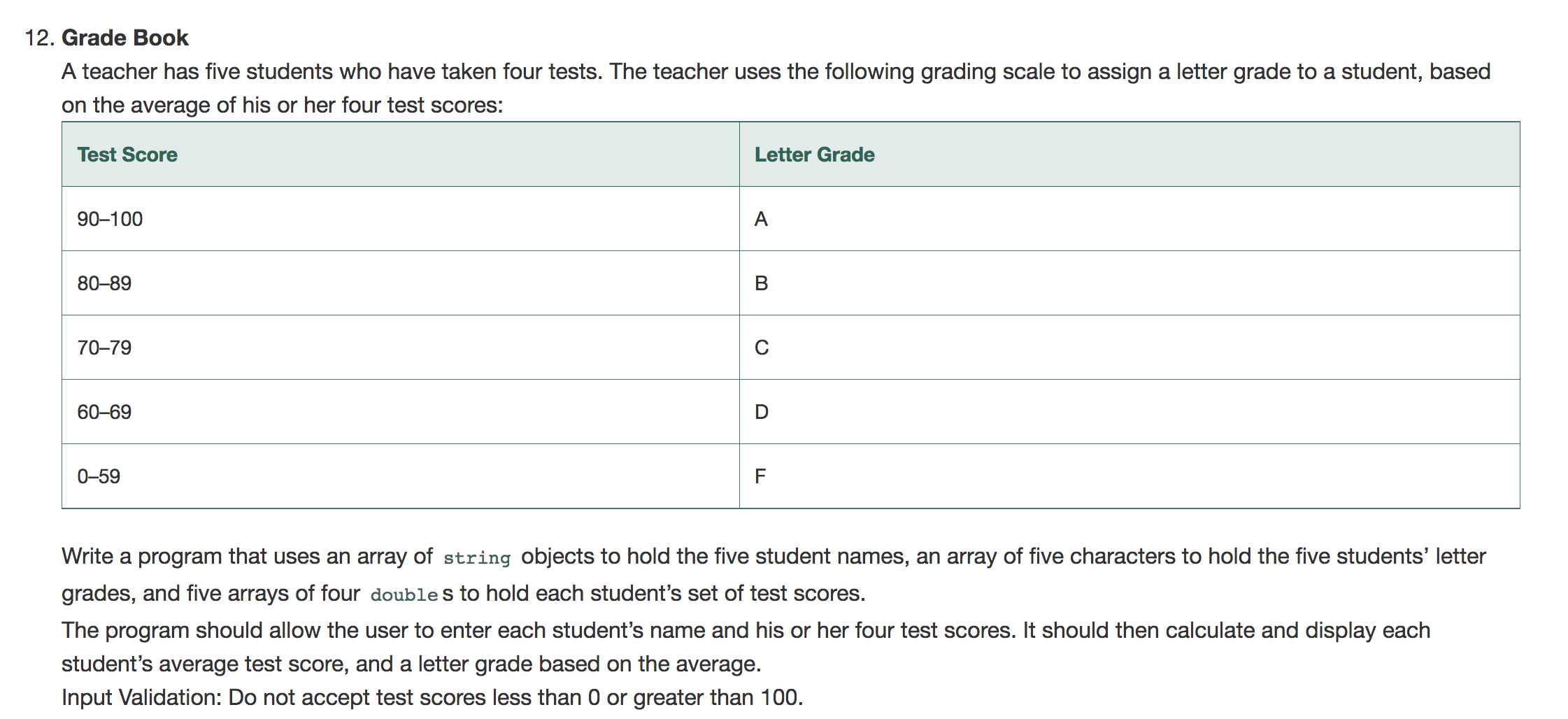
for ( int i = 0 ; i < SIZE; i++){

if ( arr[i] > n) cout << arr[i] << endl;

}

}





Answer: #include <iostream>

#include <string>

using namespace std;

double averageScore(double scores[][4], int SIZE, int student);

int main() {

const int SIZE = 5;

string name[SIZE];

char letterGrade[SIZE];

double quizes[SIZE][4];

for (int student = 0; student < SIZE; student++){

cout << "Enter Student Name: " << endl;

getline(cin, name[student], '\n');

for (int quiz = 0; quiz < 4; quiz++){

do{

cout << "Enter their grade: (range 0 - 100)" << endl;

cin >> quizes[student][quiz];

}while(quizes[student][quiz]< 0 || quizes[student][quiz] > 100);

}

cin.ignore(2, '\n');

}

for (int student = 0; student < SIZE; student++){

double average = averageScore(quizes,SIZE,student);

letterGrade[student] = average <= 59 ? 'F' :

average <= 69 ? 'D' :

average <= 79 ? 'C' :

average <= 89 ? 'B' :

average <= 100 ? 'A' : 'N';

cout << "Name: " << name[student] << " Grade: " << letterGrade[student] << endl;

}

return 0;

}

double averageScore(double scores[][4], int SIZE, int student){

double total = 0;

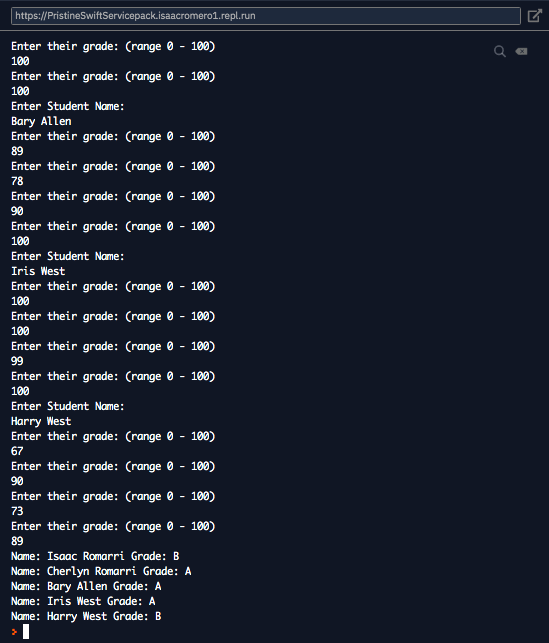
for (int testNumber = 0 ; testNumber < SIZE-1; testNumber++){

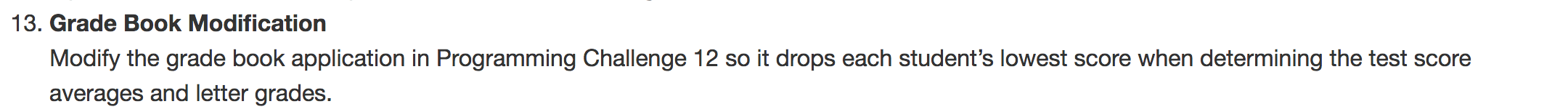
total+= scores[student][testNumber];

}

return total/(SIZE-1);

}





Answer: ( it is the same code but the averageScore function change to the following)

double averageScore(double scores[][4], int SIZE, int student){

double total = 0;

double min = scores[student][0];

for (int testNumber = 0 ; testNumber < SIZE-1; testNumber++){

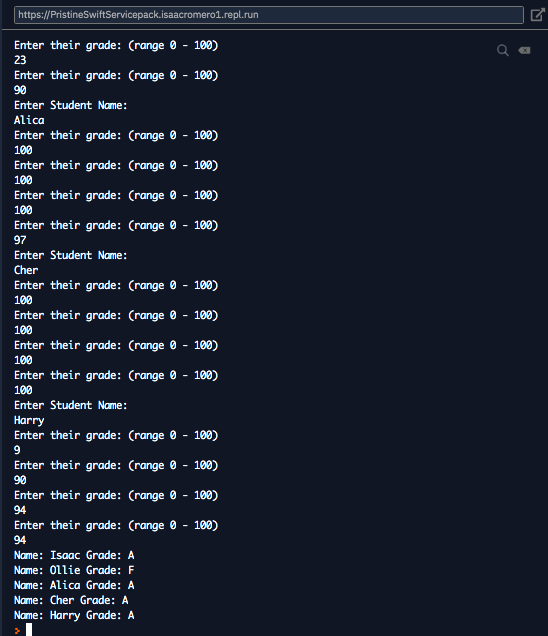
total+= scores[student][testNumber];

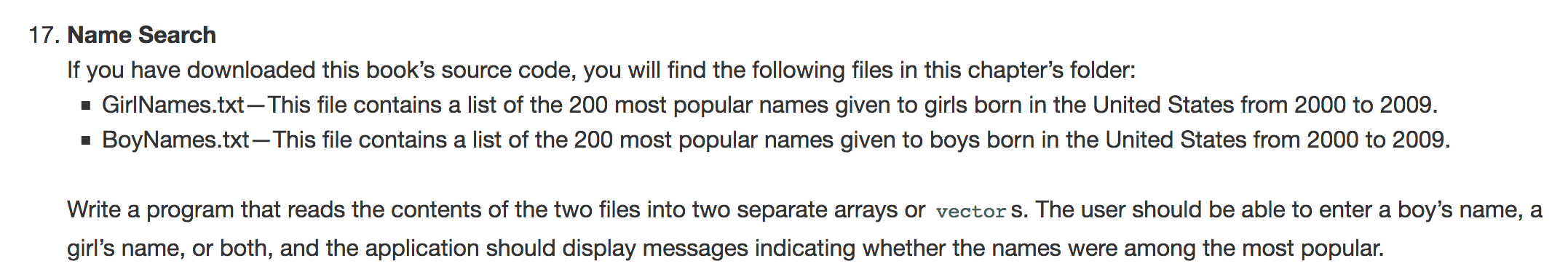
min = scores[student][testNumber] < min ? scores[student][testNumber] : min;

}

return (total-min)/(SIZE-2); // take min out of the sum

}





Answer:

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

bool search(vector<string> list, string name);

string toLower(string name);

int main() {

ifstream girlNamesFile, boyNamesFile;

girlNamesFile.open("GirlNames.txt");

boyNamesFile.open("BoyNames.txt");

vector<string> gNames;

vector<string> bNames;

string name;

while(girlNamesFile >> name){

gNames.push\_back(name);

}

while(boyNamesFile >> name){

bNames.push\_back(name);

}

string favGName;

string favBName;

cout << "Tell Me your fav girl name: " << endl;

getline(cin, favGName,'\n');

if (!search(gNames,favGName)) cout << "Sorry, Its not among the popular names. In other words, It is unique!" << endl;

cout << "Tell me your fav boy name: " << endl;

getline(cin, favBName,'\n');

if (!search(bNames,favBName)){

cout << "Sorry, it is not popular. It must be unique. :) " << endl;

}

return 0;

}

bool search(vector<string> list, string name){

bool found = false;

name = toLower(name);

for ( vector<string>::iterator it = list.begin(); it != list.end(); ++it){

if (toLower(\*it) == name){

cout << "It is popular!" << endl;

return true;

}

}

return false;

}

string toLower(string name){

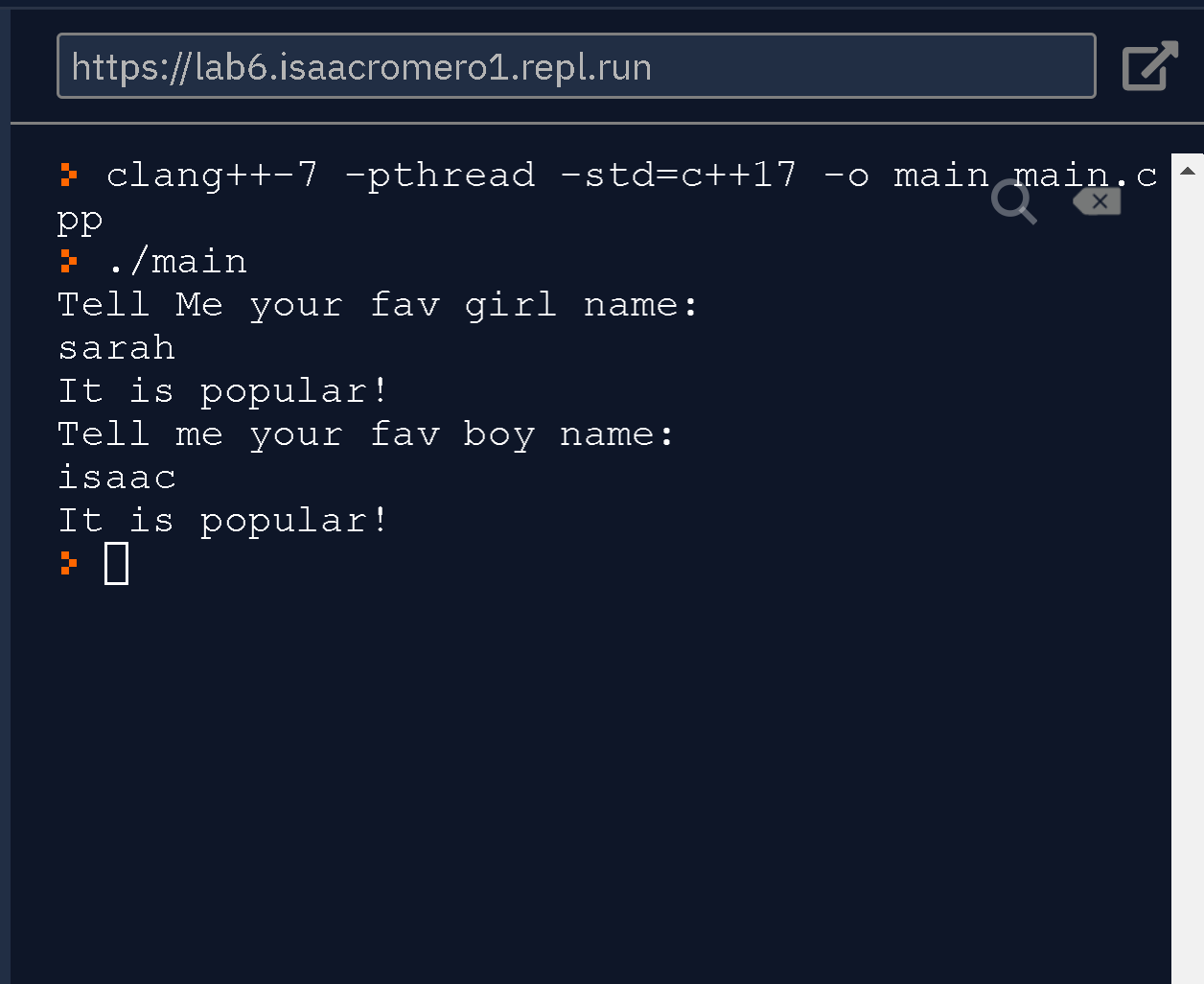
for\_each(name.begin(), name.end(), [](char & c) {

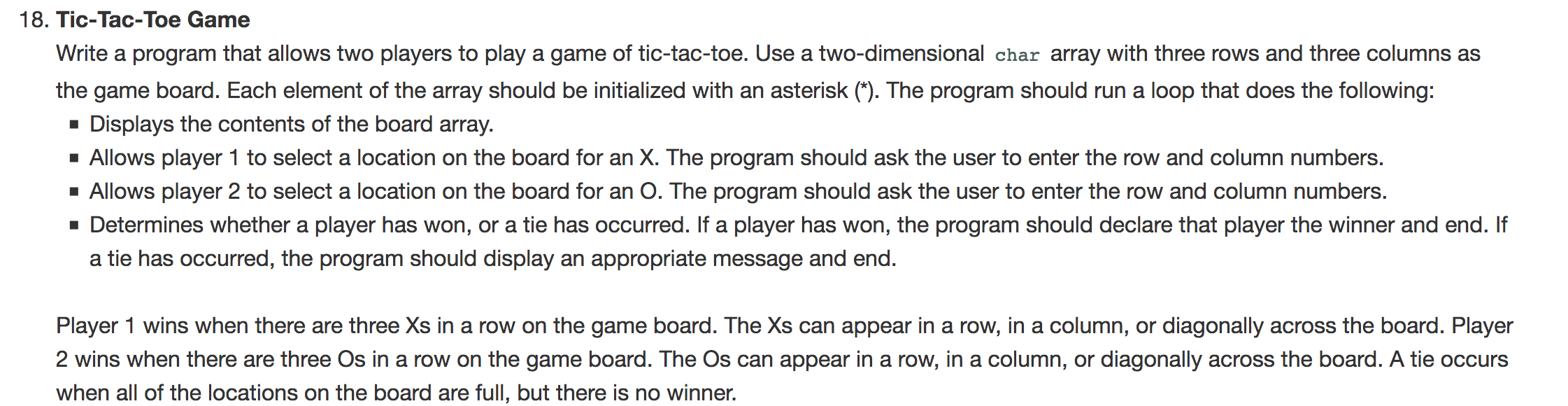
c = ::toupper(c);

});

return name;

}





Answer:

#include <iostream>

#include <string>

#include <stdlib.h>

using namespace std;

void updateBoard(int selection[9]);

string makeARow(int selection[9], int \*index);

void swapPlayers(int \*player);

bool winner(int selection[9]);

int main() {

bool gameover = false;

int player = 2; // to switch between player one and two

int selection[9]={'t','i','k','+','a','k','T','o','e'};

int option;

int rounds = 1;

do{

swapPlayers(&player);

system("clear");

cout << "Player " << player << endl;

cout << "Round " << rounds << endl;

updateBoard(selection);

cout << "Select where to place your X/O:" << endl;

cin >> option;

selection[option-1] = player;

gameover = winner(selection);

rounds++;

}while(!gameover && rounds < 9);

if (rounds == 9 ) {

string cont;

cout << "It is a draw" << endl;

getline(cin,cont);

return main();

}

system("clear");

cout << "Game Completed" << endl;

updateBoard(selection);

cout << "Congratulations Player " << player << " Won!!" << endl;

return 0;

}

bool winner(int selections[9]){

for (int i = 0; i < 9; i+=3){

// horizontal wins

if(selections[i] == selections[i+1] && selections[i+1]==selections[i+2]) return true;

// vertical wins

if(selections[i/3] == selections[i/3+3] && selections[i/3+3] == selections[i/3+6]) return true;

}

// diagonal wins

if(selections[1-1] == selections[5-1] && selections[5-1] == selections[9-1]) return true;

if(selections[3-1] == selections[5-1] && selections[5-1] == selections[7-1]) return true;

return false;

}

void swapPlayers(int \*player){

if (\*player == 1) \*player = 2;

else \*player = 1;

}

void updateBoard(int selection[9]){

int index = 1;

string middleLine = "---|---|---";

string row = "";

for(int i = 0; i < 5; i++){

switch(i){

case 1:

case 3:

cout << middleLine << endl;

break;

case 0:

case 2:

case 4:

row = makeARow(selection, &index);

cout << row << endl;

break;

default:

cout << "Something went wrong" << endl;

}

}

}

string makeARow(int selection[9],int \*index){

string row = "";

for (;;){

string spot;

if ( selection[\*index-1] == 1) spot = 'x';

else if ( selection[\*index-1] == 2) spot = 'o';

else spot = to\_string(\*index);

row += " ";

row += spot ;

row += " |";

if (\*index % 3 == 0){

\*index = \*index +1;

break;

}

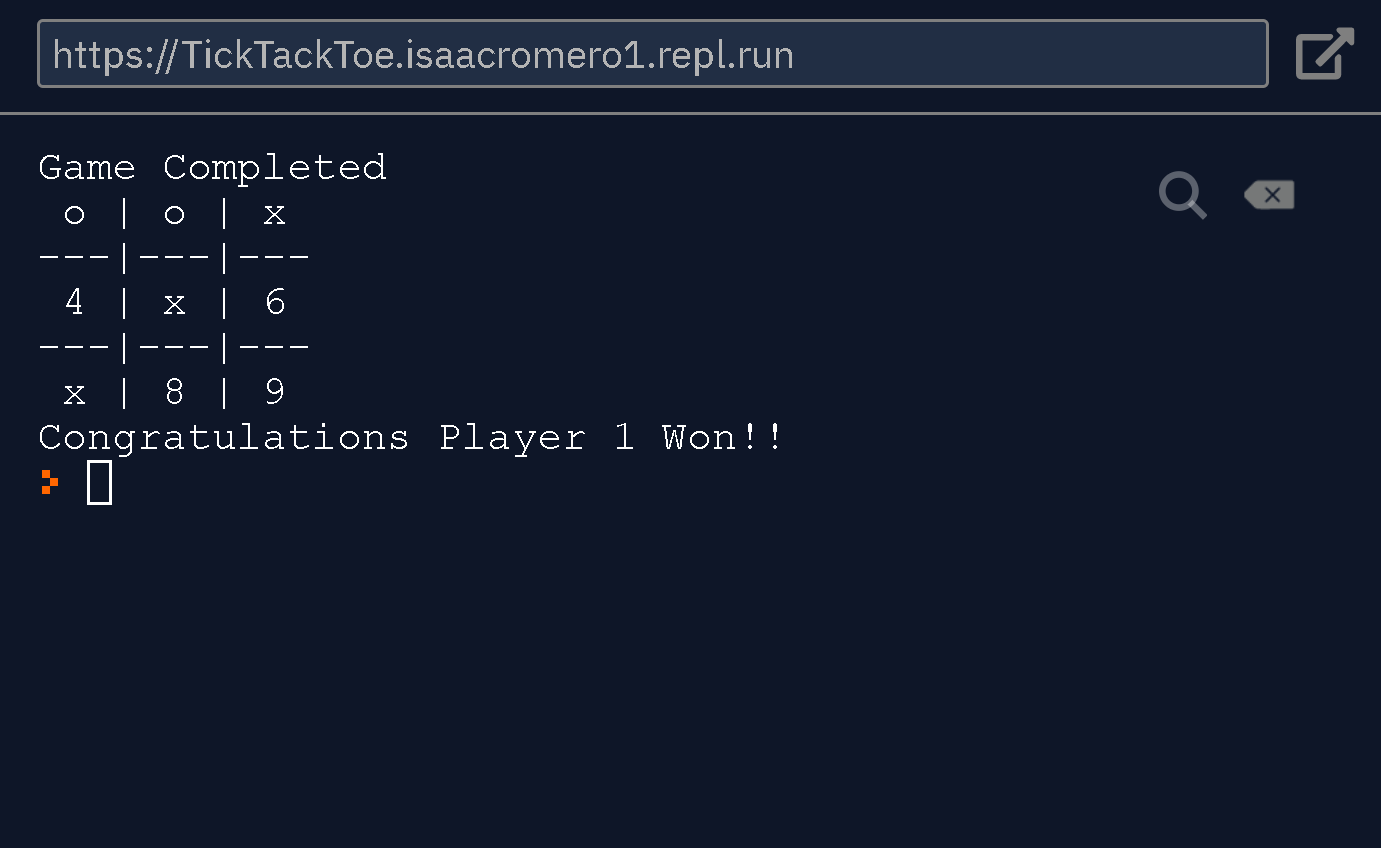
\*index = \*index + 1;

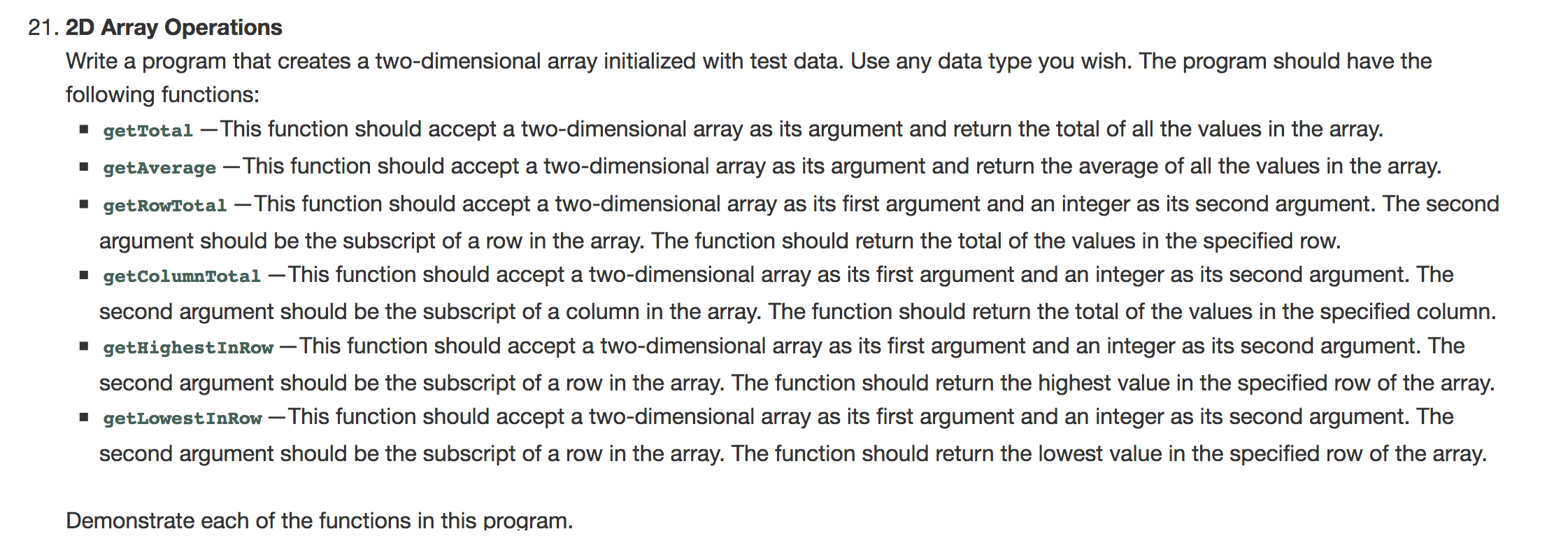
}

row.pop\_back(); // removing the last vertical line

return row;

}





Answer: