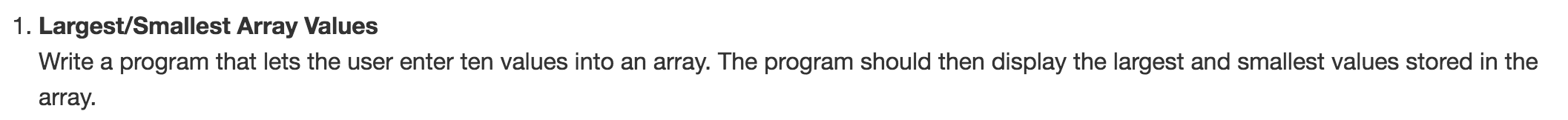
COMSC-110 Lab-6 Due: July-12-2020

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;

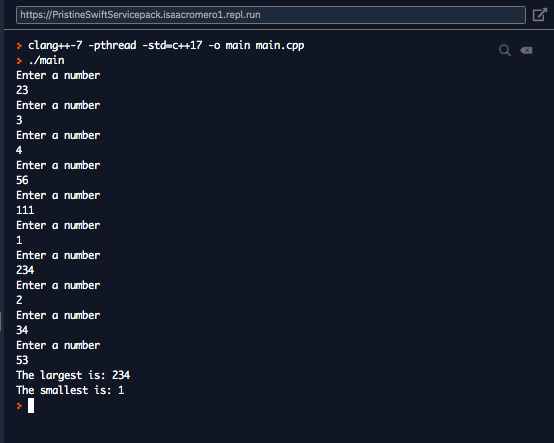
}

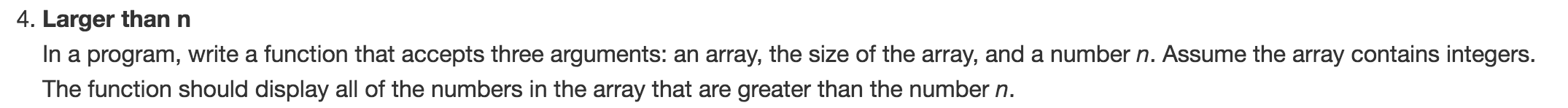
cout << "The largest is: " << max << endl;

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){

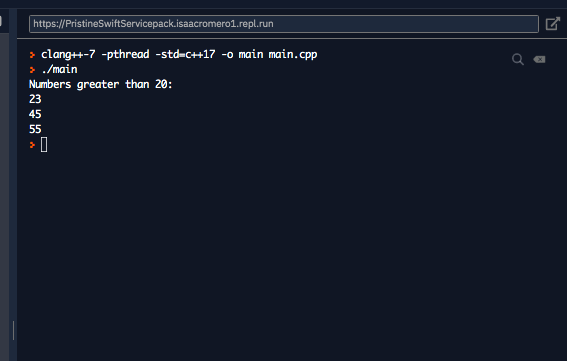
cout << "Numbers greater than " << n << ":" << endl;

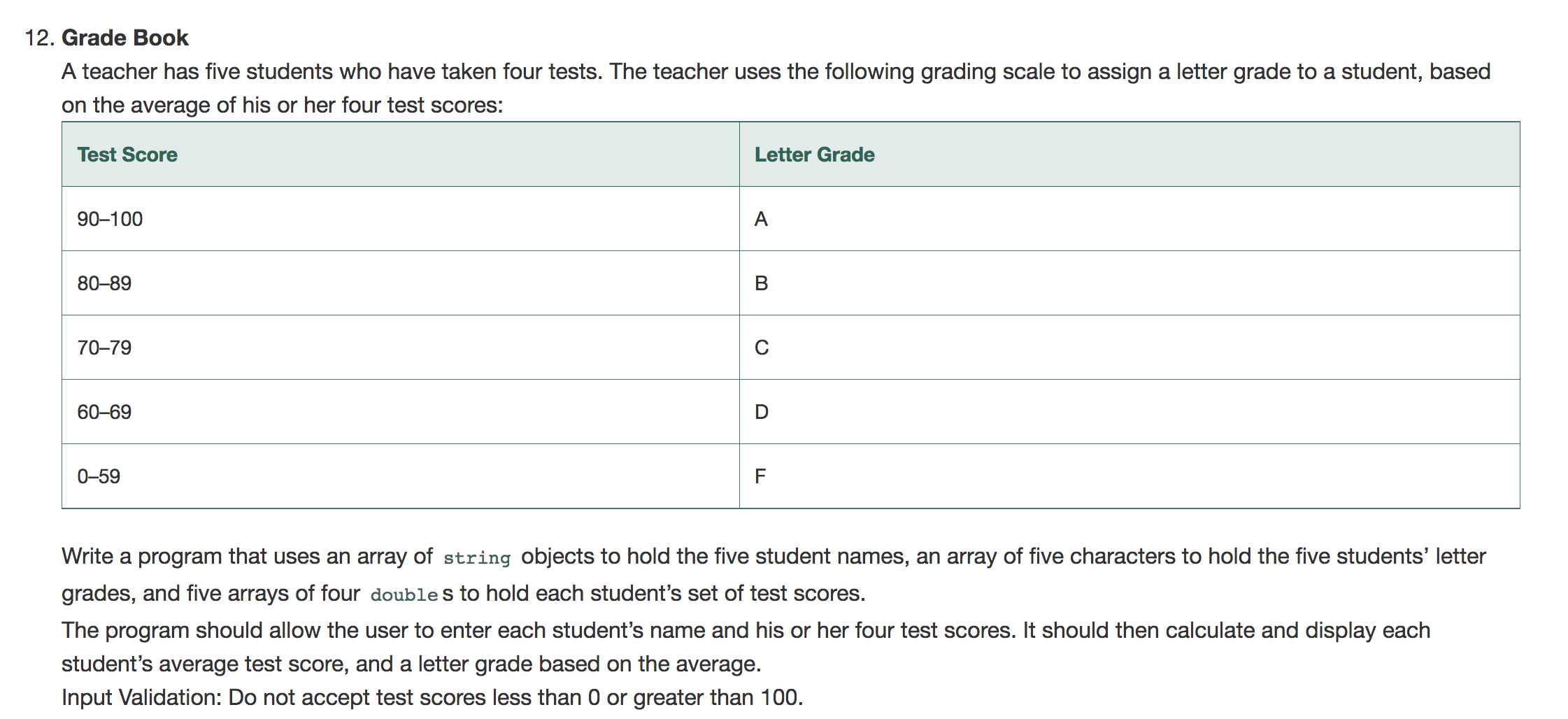
for ( int i = 0 ; i < SIZE; i++){

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;

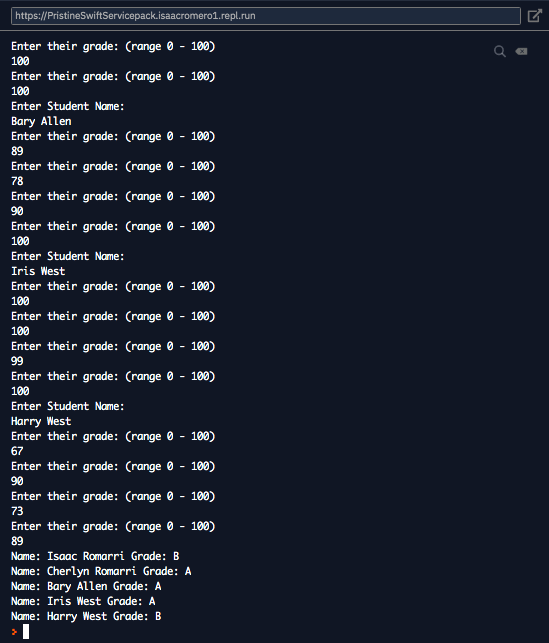
for (int testNumber = 0 ; testNumber < SIZE-1; testNumber++){

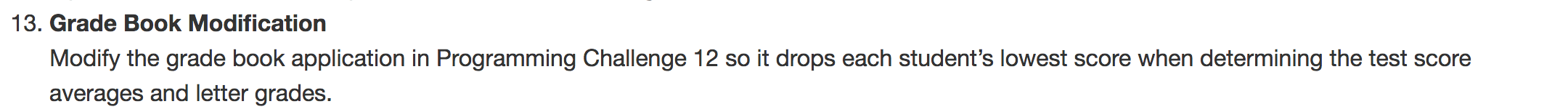
total+= scores[student][testNumber];

}

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++){

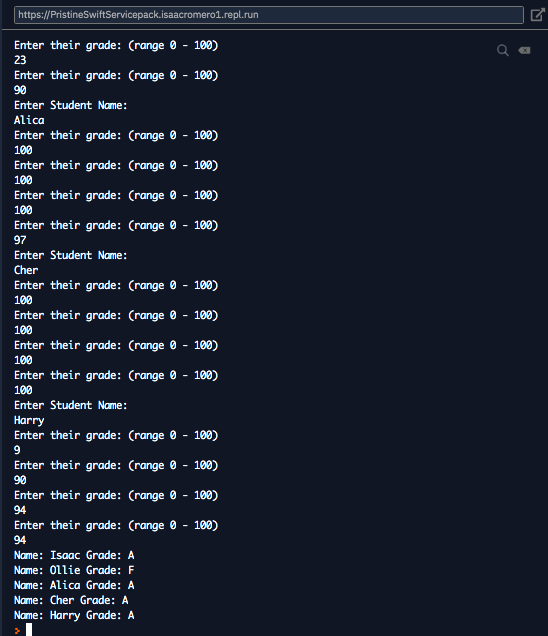
total+= scores[student][testNumber];

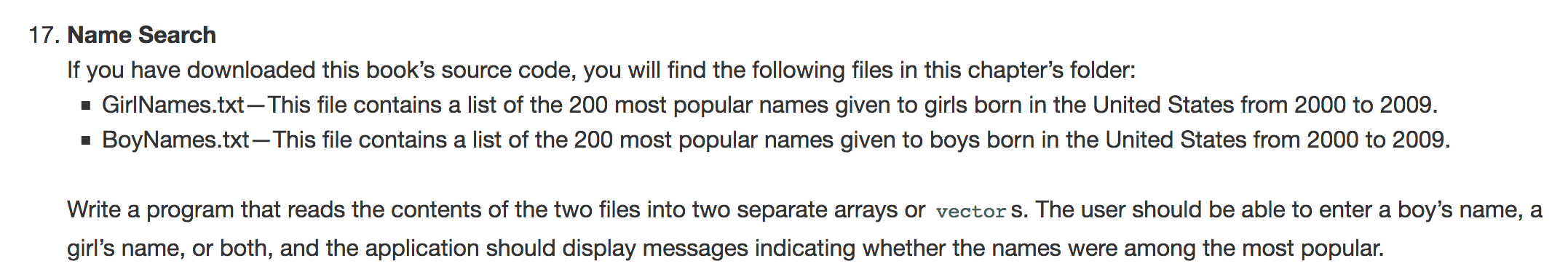
min = scores[student][testNumber] < min ? scores[student][testNumber] : min;

}

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){

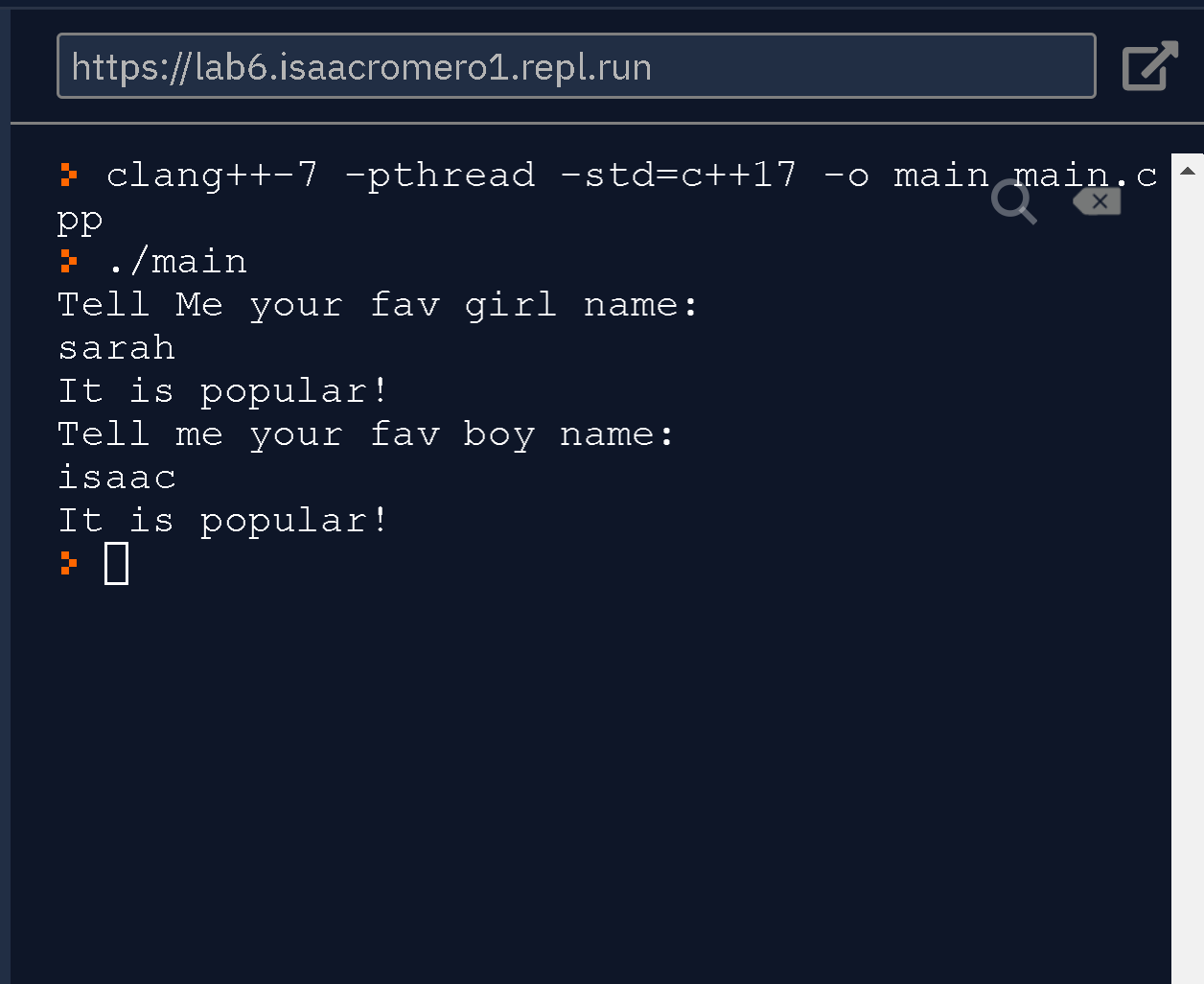
for\_each(name.begin(), name.end(), [](char & c) {

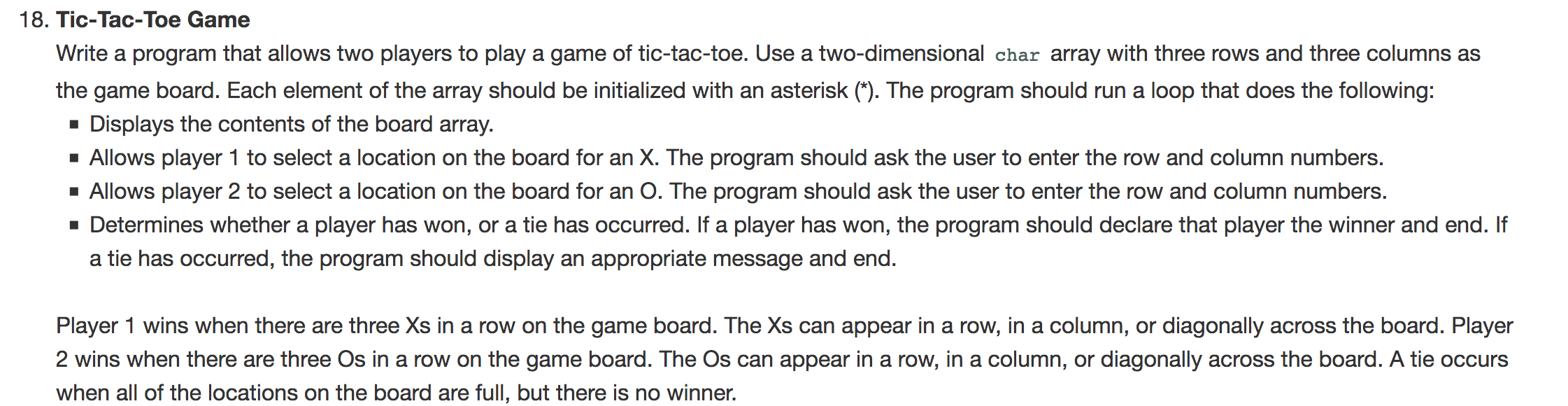
c = ::toupper(c);

});

return name;

}





Answer:

#include <iostream>

using namespace std;

void display(char[][3], int);

void fill(char[][3], int);

bool swapPlayers(bool);

void updateBoard(char[][3], int, int, int);

bool winner(char[][3], int);

bool validateUserInput(char[][3], int, int, int);

int main() {

bool gameOver = false;

int round = 1;

const int SIZE = 3;

char board[SIZE][SIZE];

fill(board,SIZE); // initializing board

cout << "Round: " << round << endl;

display(board,SIZE); // Displays the board

int row, col;

while(!gameOver){

do{

cout << "Enter row ,and col (1 2): " << endl;

cin >> row >> col;

}while(validateUserInput(board,SIZE,row-1,col-1));

updateBoard(board, SIZE, row-1, col-1);

system("clear");

cout << "Round: " << round << endl;

display(board, SIZE);

gameOver = winner(board, SIZE);

if (round > 9) gameOver = true;

round++;

swapPlayers(true);

}

if (gameOver && round != 9){

cout << "Congrats! Player " << swapPlayers(true) << endl;

}

}

bool validateUserInput(char board[][3], int SIZE, int row, int col){

char player = !swapPlayers(false)? 'x' : 'o';

if (row > 3 || row < 0 || col <0 || col > 3){

cout << "Not a good range, try '1 1' " << endl;

return true;

}

if (board[row][col] != '\*'){

cout << "Player " << player << " has that spot!" << endl;

return true;

}

return false;

}

bool winner(char board[][3], int SIZE){

char player = swapPlayers(false) ? 'x' : 'o';

int counter = 0;

// check horizontal wins

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

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

if ( board[row][col] == player) counter++;

}

if (counter == 3) return true;

else counter = 0;

}

// check vertical wins

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

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

if ( board[row][col] == player) counter++;

}

if (counter == 3) return true;

else counter = 0;

}

// check diagonal wins

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

if( board[i][i] == player) counter++;

}

if (counter == 3) return true;

else counter = 0;

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

return true;

}

return false;

}

void updateBoard(char board[][3],int SIZE, int row, int col ){

board[row][col] = swapPlayers(false) ? 'x' : 'o';

}

bool swapPlayers(bool swap){

static bool player = true;

if (swap) player = !player;

return player;

}

void display(char board[][3], int SIZE){

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

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

cout << board[row][col] << '\t' ;

}

cout << endl;

}

}

void fill(char board[][3], int SIZE){

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

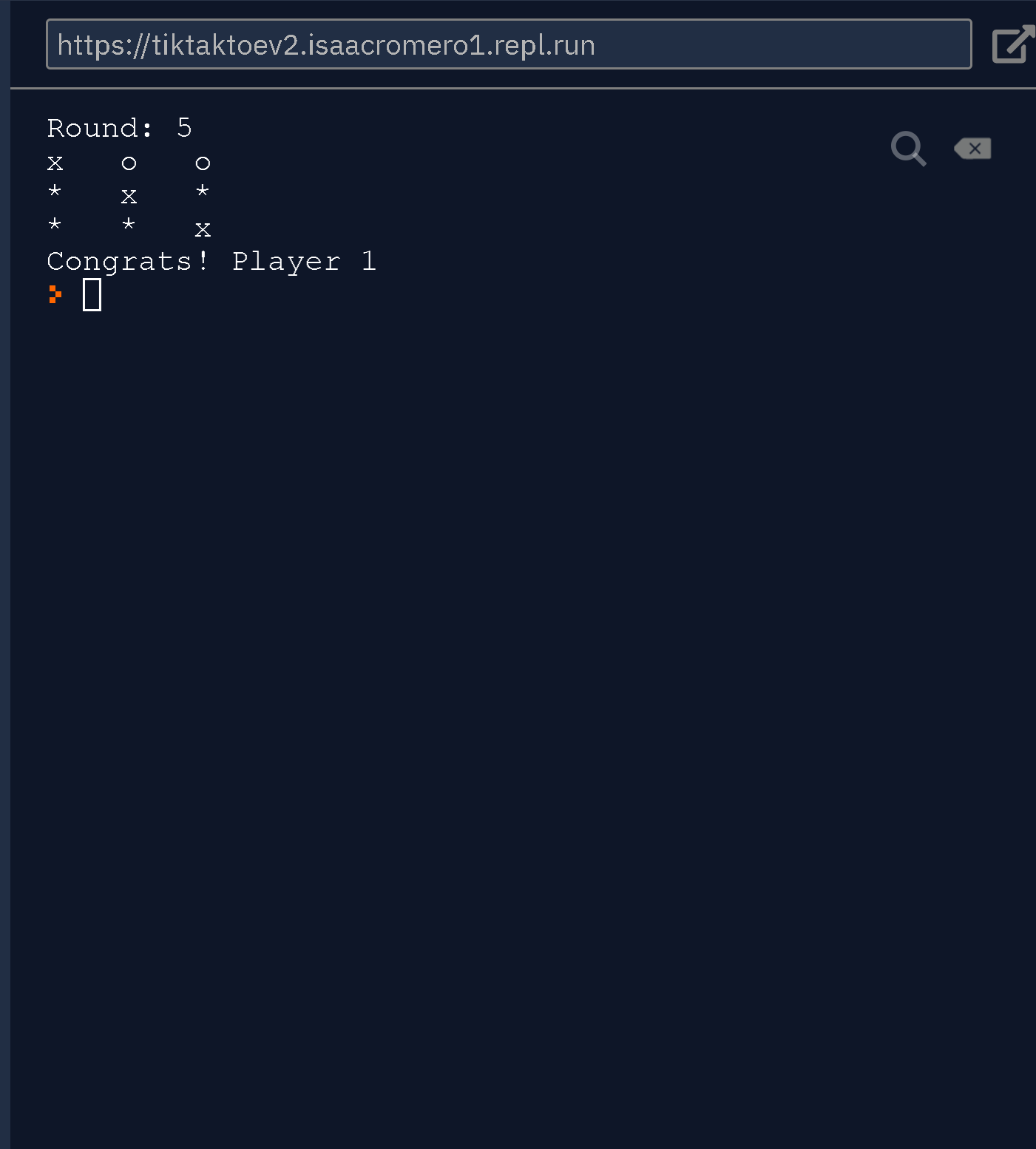
for (int col = 0; col < SIZE; col++){

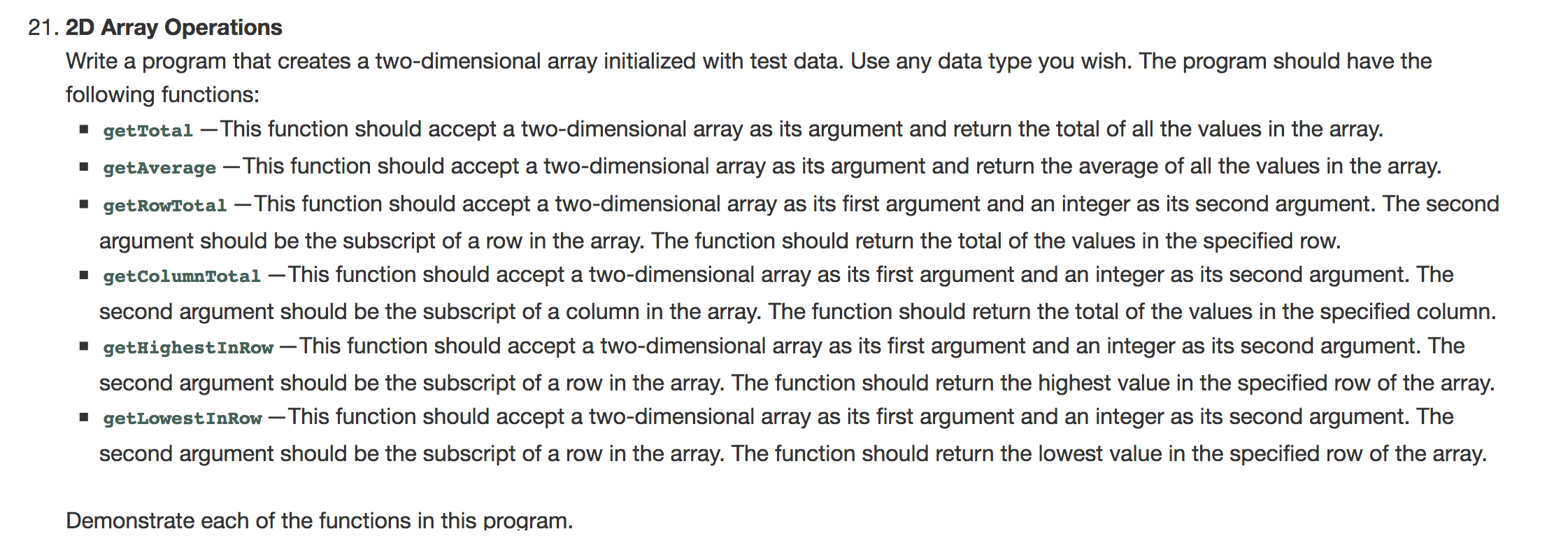
board[row][col] = '\*';

}

}

}





Answer:

#include <iostream>

#include <stdlib.h>

using namespace std;

void display(int \*);

void fillArray(int \*);

void foreach(int ar[][4], int SIZE, void (\*f)(int \*));

int getTotal(int [][4], int);

double getAverage(int [][4],int);

int getRowTotal(int [][4],int , int );

int getColTotal(int [][4], int, int);

int getLowestInRow(int [][4], int, int);

int getHighestInRow(int [][4], int, int);

int main() {

srand(time(NULL));

const int SIZE = 4;

int ar[SIZE][SIZE] = {0};

foreach(ar, SIZE, fillArray);

foreach(ar, SIZE, display);

cout << "Row 0 Total: " << endl;

cout << getRowTotal(ar,SIZE,0) << endl;

cout << "Col 1 Total: " << endl;

cout << getColTotal(ar,SIZE,1) << endl;

cout << "Total: " << getTotal(ar, SIZE) << endl;

cout << "Average: " << getAverage(ar,SIZE) << endl;

cout << "The lowest in row 0 is: " << getLowestInRow(ar,SIZE,0) << endl;

cout << "The highest in row 0 is " << getHighestInRow(ar, SIZE, 0) << endl;

return 0;

}

int getHighestInRow(int ar[][4], int SIZE, int row){

int max = ar[row][0];

for ( int col = 1; col < SIZE; col++){

if(ar[row][col] > max) max = ar[row][col];

}

return max;

}

int getLowestInRow(int ar[][4], int SIZE, int row){

int min = ar[row][0];

for ( int col = 1; col < SIZE; col++){

if(ar[row][col] < min) min = ar[row][col];

}

return min;

}

int getRowTotal(int ar[][4],int SIZE, int row){

int total = 0;

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

total+= ar[row][col];

}

return total;

}

int getColTotal(int ar[][4], int SIZE, int col){

int total = 0;

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

total+= ar[row][col];

}

return total;

}

int getTotal(int ar[][4], int SIZE){

int total = 0;

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

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

total+= ar[row][col];

}

}

return total;

}

double getAverage(int ar[][4], int SIZE){

int total = 0;

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

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

total+= ar[row][col];

}

}

return static\_cast<double>(total)/SIZE/SIZE;

}

void foreach(int ar[][4], int SIZE, void (\*f)(int \*)){

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

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

(\*f)(&ar[row][col]);

}

}

}

void display(int \*number){

static int col = 0;

cout << \*number << "\t" ;

if ( col == 3){

col = 0;

cout << endl;

}else col++;

}

void fillArray(int \*num){

\*num = rand()%10;

}

