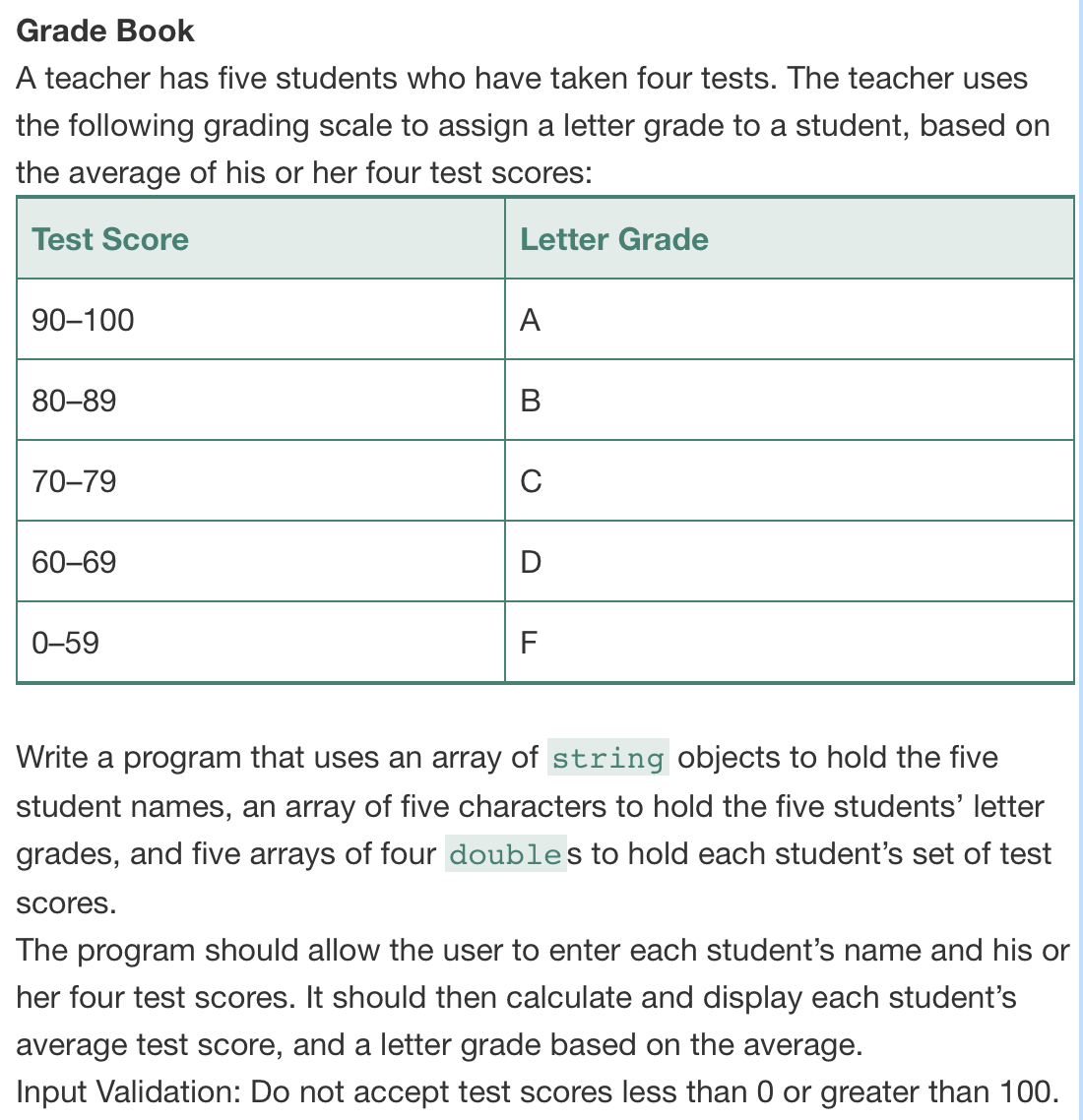
COMSC165 Due: June-24-2018 11:59 PM Name: Valentino Lei’a

If you are using a MAC, COMMAND CONTROL SHIFT 4 can be used for screen shot.

If you use a PC, snipping tools can be used for screen shot.

If you have any questions about the lab please send the question to [answerneededsoon@gmail.com](mailto:answerneededsoon@gmail.com)



#ifndef STUDENT\_H

#define STUDENT\_H

#include <string>

class Student {

public:

static const int numberOfStudents = 5;

static const int numberOfTests = 4;

static const int maxTestScore = 100;

static const int minTestScore = 0;

std::string studentNames[numberOfStudents];

char studentLetterGrade[numberOfStudents];

double studentTestScore[numberOfStudents][numberOfTests];

double averageTestScores[numberOfStudents];

void getStudentData();

void calculateAverage();

void displayStudentData();

char getStudentGrade(double);

};

#endif // STUDENT\_H

#include "Student.h"

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void Student::getStudentData(){

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

cout << "\nEnter student " << i+1 << "'s name: ";

cin >> studentNames[i];

cout << "Enter their 4 test scores: ";

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

cin >> studentTestScore[i][j];

while (studentTestScore[i][j] > maxTestScore || studentTestScore[i][j] < minTestScore){

cout << "\nInvalid score. Test score needs to be between 0 and 100." << endl;

cout << "Re-enter test score: ";

cin >> studentTestScore[i][j];

}

}

}

}

void Student::calculateAverage(){

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

averageTestScores[i] = 0;

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

averageTestScores[i] += studentTestScore[i][j];

cout << "\ndebug avgscoreindex " << i << ": " << averageTestScores[i];

}

averageTestScores[i] /= numberOfTests;

studentLetterGrade[i] = getStudentGrade(averageTestScores[i]);

}

}

void Student::displayStudentData(){

cout << "\nStudent Name\tAverage Score\tLetter Grade" << endl;

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

cout.setf(ios::fixed, ios::floatfield);

cout.precision(2);

cout << studentNames[i] << "\t\t" << setprecision(2) << averageTestScores[i] << "\t\t" << studentLetterGrade[i] << endl;

}

}

char Student::getStudentGrade(double x){

if(x >= 90 && x <= maxTestScore){

return 'A';

}else if(x >= 80 && x < 90){

return 'B';

}else if(x >= 70 && x < 80){

return 'C';

}else if(x >= 60 && x < 70){

return 'D';

}else{

return 'F';

}

}

#include <iostream>

#include "Student.h"

using namespace std;

int main()

{

Student so;

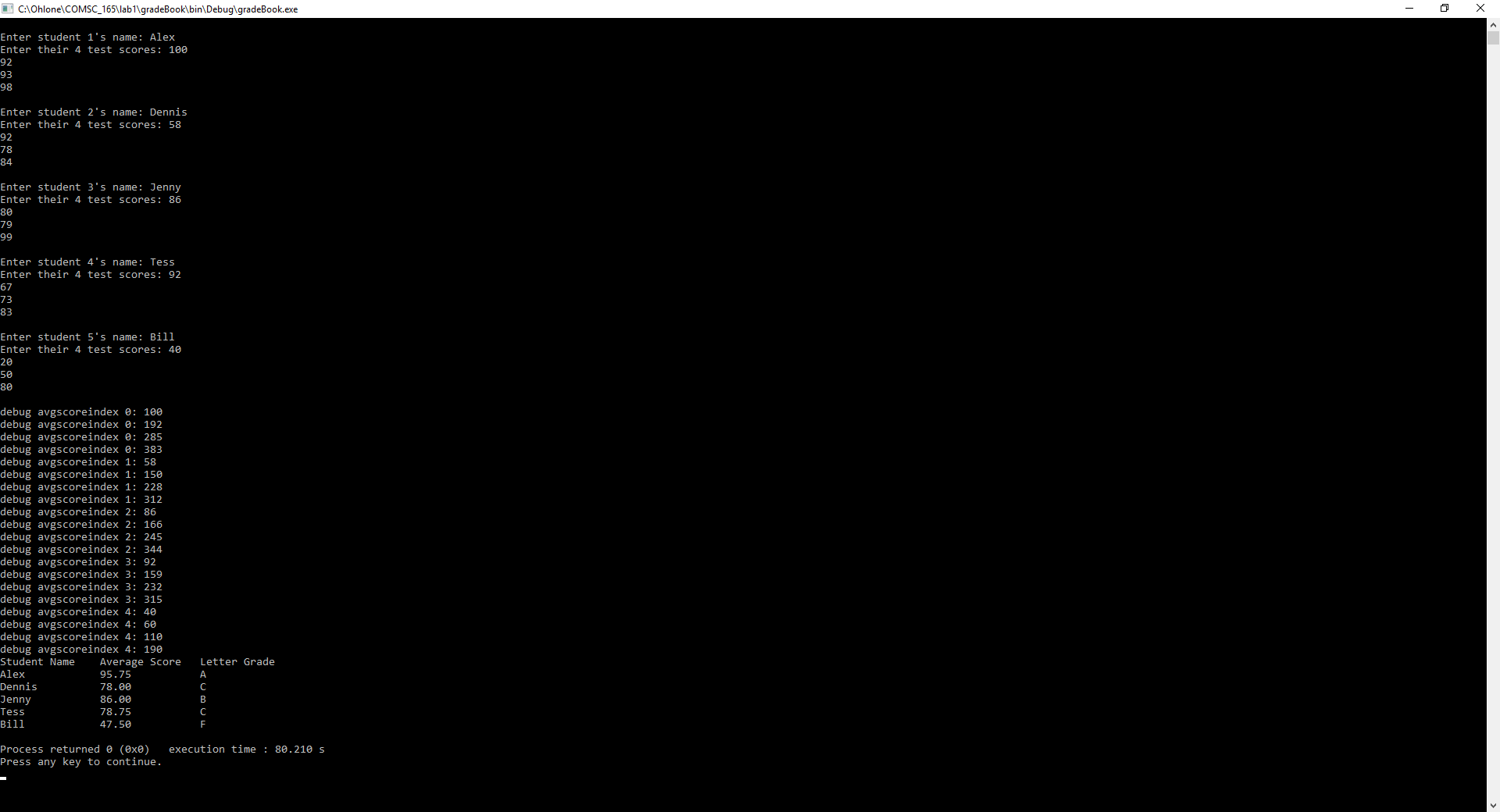
so.getStudentData();

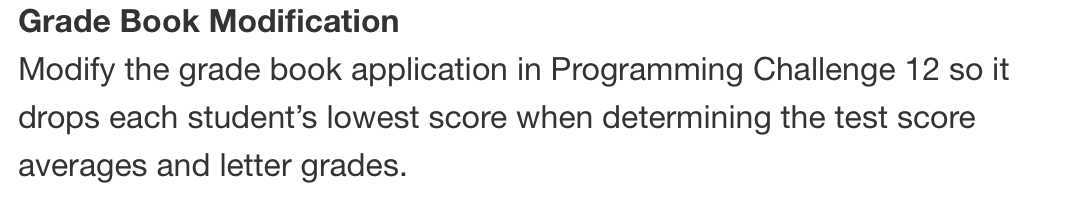
so.calculateAverage();

so.displayStudentData();

return 0;

}





#ifndef STUDENT\_H

#define STUDENT\_H

#include <string>

class Student {

public:

static const int numberOfStudents = 5;

static const int numberOfTests = 4;

static const int maxTestScore = 100;

static const int minTestScore = 0;

std::string studentNames[numberOfStudents];

char studentLetterGrade[numberOfStudents];

double studentTestScore[numberOfStudents][numberOfTests];

double averageTestScores[numberOfStudents];

void getStudentData();

void findLowestScoreAndCalculateAverage();

void displayStudentData();

char getStudentGrade(double);

};

#endif // STUDENT\_H

#include "Student.h"

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void Student::getStudentData(){

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

cout << "\nEnter student " << i+1 << "'s name: ";

cin >> studentNames[i];

cout << "Enter their 4 test scores: ";

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

cin >> studentTestScore[i][j];

while (studentTestScore[i][j] > maxTestScore || studentTestScore[i][j] < minTestScore){

cout << "\nInvalid score. Test score needs to be between 0 and 100." << endl;

cout << "Re-enter test score: ";

cin >> studentTestScore[i][j];

}

}

}

}

void Student::findLowestScoreAndCalculateAverage(){

int lowestScore = 0;

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

lowestScore = studentTestScore[i][0];

averageTestScores[i] = 0;

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

if(lowestScore >= studentTestScore[i][j]){

lowestScore = studentTestScore[i][j];

averageTestScores[i] += studentTestScore[i][j];

}

}

averageTestScores[i] -= lowestScore;

cout << "\ndebug avgscoreindex " << i << ": " << averageTestScores[i];

averageTestScores[i] /= 3;

studentLetterGrade[i] = getStudentGrade(averageTestScores[i]);

}

}

void Student::displayStudentData(){

cout << "\nStudent Name\tAverage Score\tLetter Grade" << endl;

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

cout.setf(ios::fixed, ios::floatfield);

cout.precision(2);

cout << studentNames[i] << "\t\t" << setprecision(2) << averageTestScores[i] << "\t\t" << studentLetterGrade[i] << endl;

}

}

char Student::getStudentGrade(double x){

if(x >= 90 && x <= maxTestScore){

return 'A';

}else if(x >= 80 && x < 90){

return 'B';

}else if(x >= 70 && x < 80){

return 'C';

}else if(x >= 60 && x < 70){

return 'D';

}else{

return 'F';

}

}

#include <iostream>

#include "Student.h"

using namespace std;

int main()

{

Student so;

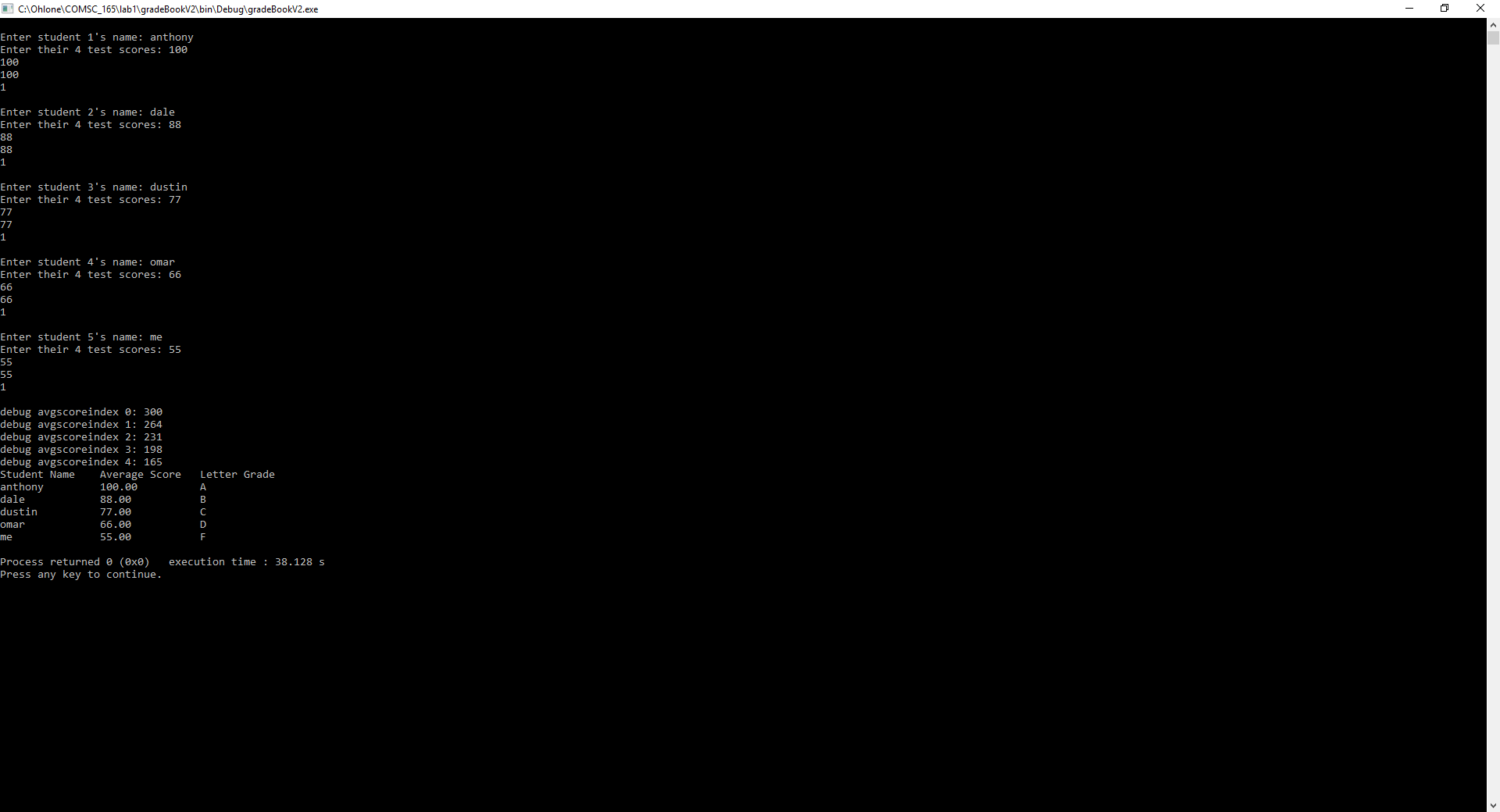
so.getStudentData();

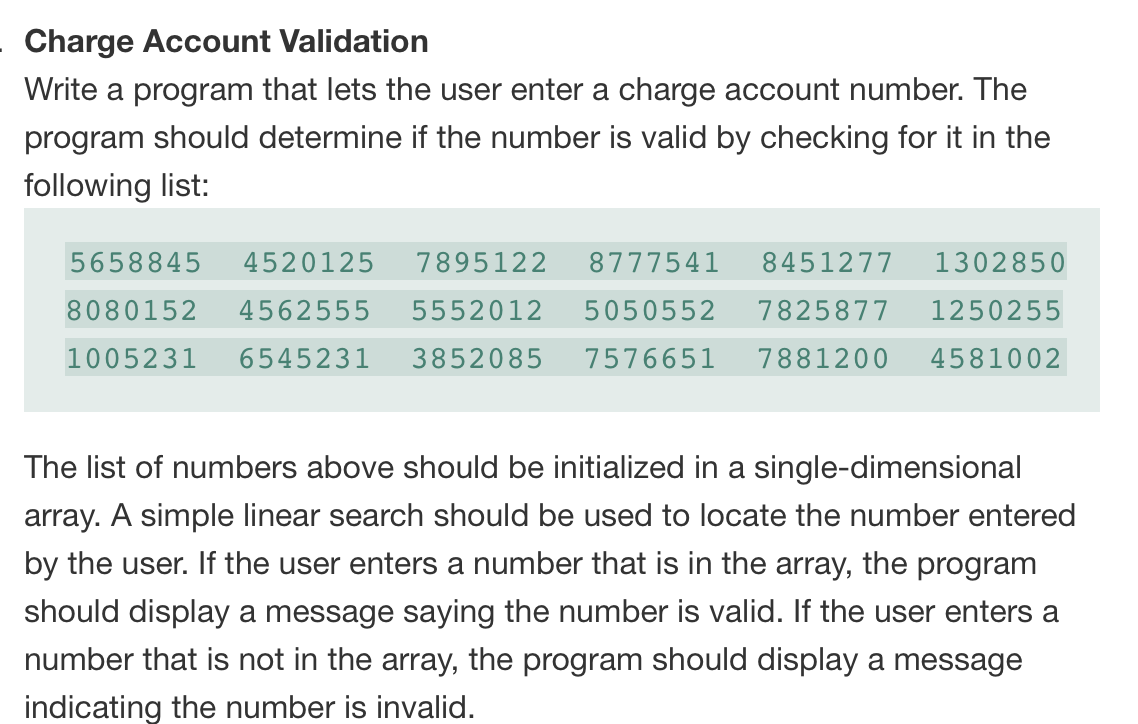
so.findLowestScoreAndCalculateAverage();

so.displayStudentData();

return 0;

}





#ifndef ACCOUNT\_H

#define ACCOUNT\_H

class Account

{

public:

static const int arraySize = 18;

void linearArraySearchAndDisplayResults(int[], int, int); // search array elements and display results

};

#endif // ACCOUNT\_H

#include "Account.h"

#include <iostream>

using namespace std;

void Account::linearArraySearchAndDisplayResults(int validChargeNumbers[], int arraySize, int accountChargerNumber){

bool chargeNumberMatch = false;

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

if(accountChargerNumber == validChargeNumbers[i]){

chargeNumberMatch = true;

break;

}

}

if(chargeNumberMatch = true){

cout << "\nNumber is valid.";

}else{

cout << "\nNumber is invalid.";

}

}

#include <iostream>

#include "Account.h"

using namespace std;

int main()

{

int accountChargeNumber = 0;

int validChargeNumbers[18] = {5658845, 4520125, 7895122, 8080152, 4562555, 5552012,

5050552, 7825877, 1250255, 8777541, 8451277, 1302850,

1005231, 6545231, 3852085, 7576651, 7881200, 4581002};

cout << "Enter account charge number: ";

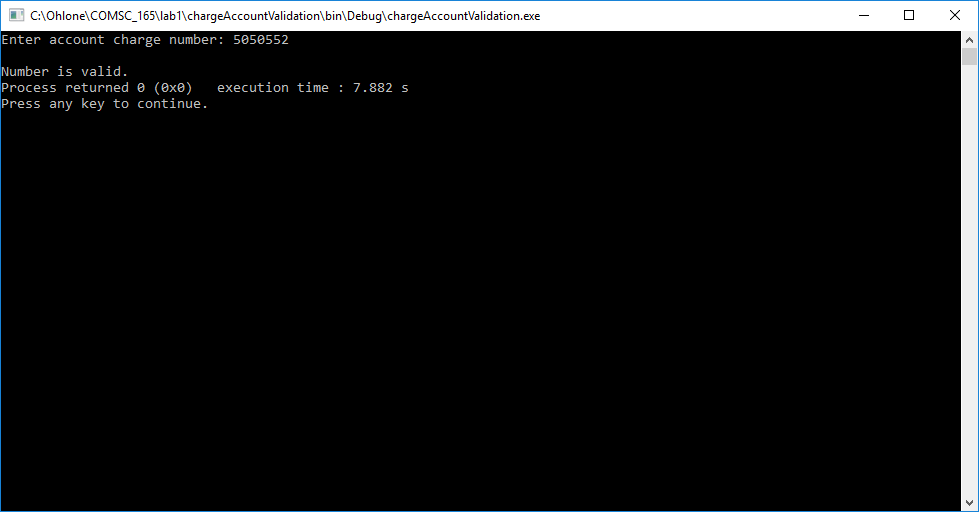
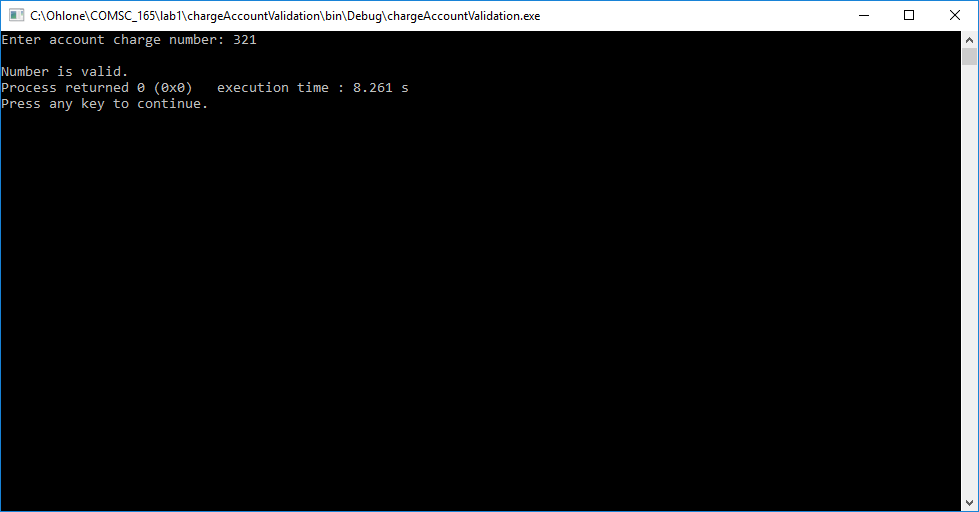
cin >> accountChargeNumber;

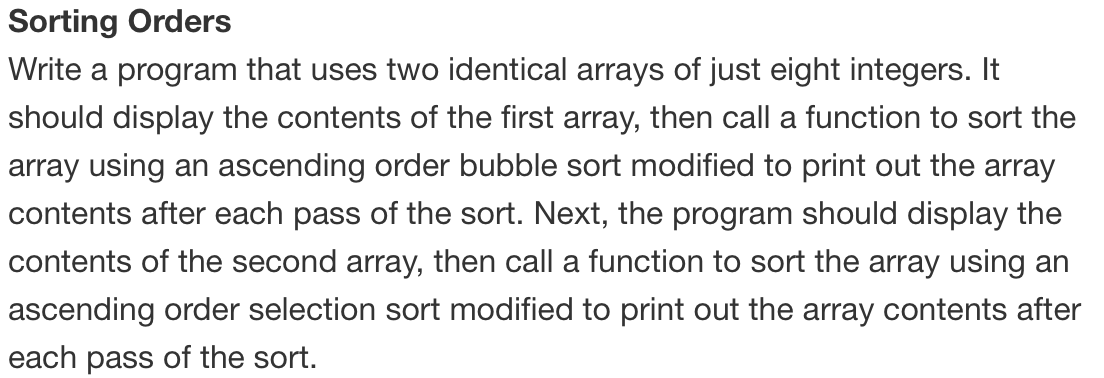
Account ao;

ao.linearArraySearchAndDisplayResults(validChargeNumbers, ao.arraySize, accountChargeNumber);

return 0;

}





#ifndef SORTORDER\_H

#define SORTORDER\_H

class SortOrder

{

public:

const int arraySize = 10;

void bubbleSort(int[], int);

void selectionSort(int[], int);

protected:

private:

};

#endif // SORTORDER\_H

#include "SortOrder.h"

#include <iostream>

using namespace std;

void SortOrder::bubbleSort(int unsortedArray[], int arraySize){

int maxElement, index;

cout << "Unsorted array: ";

for(index = 0; index < arraySize; index++){

cout << unsortedArray[index] << " ";

}

for(maxElement = arraySize - 1; maxElement > 0; maxElement--){

for(index = 0; index < maxElement; index++){

if( unsortedArray[index] > unsortedArray[index + 1]){

swap(unsortedArray[index], unsortedArray[index + 1]);

}

}

}

cout << "\nSorted array: ";

for(index = 0; index < arraySize; index++){

cout << unsortedArray[index] << " ";

}

}

void SortOrder::selectionSort(int unsortedArray[], int arraySize){

int minIndex, minValue, start, index;

cout << "\nUnsorted array: ";

for(index = 0; index < arraySize; index++){

cout << unsortedArray[index] << " ";

}

for(start = 0; start < arraySize - 1; start++){

minIndex = start;

minValue = unsortedArray[start];

for(index = start + 1; index < arraySize; index++){

if(unsortedArray[index] < minValue){

minValue = unsortedArray[index];

minIndex = index;

}

}

swap(unsortedArray[minIndex], unsortedArray[start]);

}

cout << "\nSorted array: ";

for(index = 0; index < arraySize; index++){

cout << unsortedArray[index] << " ";

}

}

#include <iostream>

#include "SortOrder.h"

using namespace std;

int main()

{

SortOrder so;

int arrayOne[so.arraySize] = {8,4,5,1,3,9,10,2,6,7};

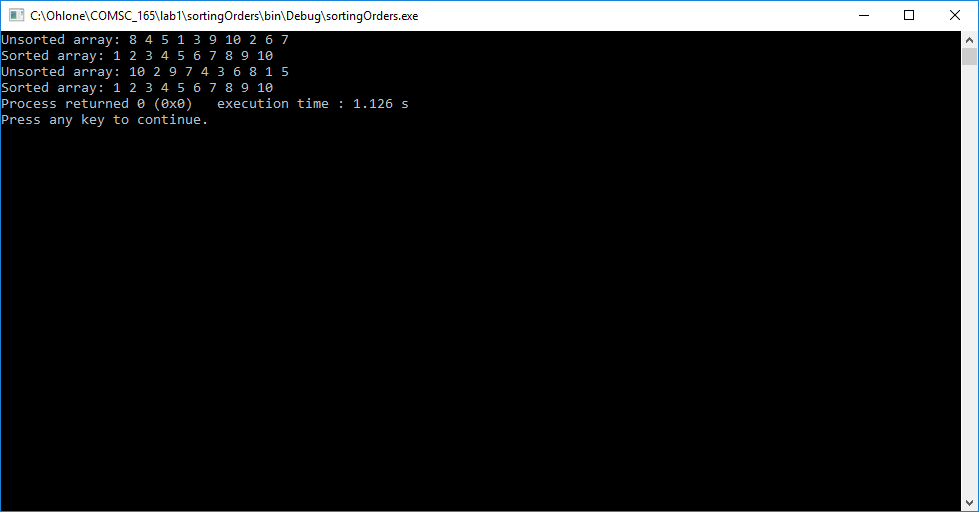
int arrayTwo[so.arraySize] = {10,2,9,7,4,3,6,8,1,5};

so.bubbleSort(arrayOne, so.arraySize);

so.selectionSort(arrayTwo, so.arraySize);

return 0;

}



1. Convert the following binary numbers to HEX. Show your work!
2. 0011 | 0101 | 0010 = 354 (hex)
3. 0011 | 1110 | 0101 | 1110 = 3E5E (hex)
4. 0101 | 0111 = 75 (hex)
5. 0010 = 2 (hex)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) |  | **0011** | **0101** | **0100** |
|  |  | 0^8 + 0^4 + 1^2 + 1^1 | 0^8 + 1^4 + 0^2 + 1^1 | 0^8 + 1^4 + 0^2 + 0^1 |
|  |  | 0 + 0 + 2 + 1 = 3 | 0 + 4 + 0 + 1 = 5 | 0 + 4 + 0 + 0 = 4 |
|  |  | 3 | 5 | 4 |
| b) | **0011** | **1110** | **0101** | **1110** |
|  | 0^8 + 0^4 + 1^2 + 1^1 | 1^8 + 1^4 + 1^2 + 0^1 | 0^8 + 1^4 + 0^2 + 1^1 | 1^8 + 1^4 + 1^2 + 0^1 |
|  | 0 + 0 + 2 + 1 = 3 | 8 + 4 + 2 + 0 = 14 | 0 + 4 + 0 + 1 = 5 | 8 + 4 + 2 + 0 = 14 |
|  | 3 | E | 5 | E |
| c) |  |  | **0101** | **0111** |
|  |  |  | 0^8 + 1^4 + 0^2 + 1^1 | 0^8 + 1^4 + 1^2 + 1^1 |
|  |  |  | 0 + 4 + 0 + 1 = 5 | 0 + 4 + 2 + 1 = 7 |
|  |  |  | 5 | 7 |
| d) |  |  |  | **0010** |
|  |  |  |  | 0^8 + 0^4 + 1^2 + 0^1 |
|  |  |  |  | 0 + 0 + 2 + 0 = 2 |
|  |  |  |  | 2 |

1. A computer has 38 address lines. How many locations in Mega bytes can it address?

262,144 MB

1. What are the ASCII values for characters A, B, C, a, b and c? What is the relationship between ASCII values of uppercase values and lowercase values?

The difference is 32.

|  |  |
| --- | --- |
| ASCII value | Decimal Value |
| A | 65 |
| B | 66 |
| C | 67 |
| a | 97 |
| b | 98 |
| c | 99 |

1. Write a program that reads two integer numbers from the keyboard (use cin). Add the two numbers and display the sum.

#include <iostream>

using namespace std;

int main(){

int x, y, sum;

cout << "Enter two integers: ";

cin >> x;

cin >> y;

sum = x + y;

cout << "sum: " << sum;

}

1. Write a program to read 4 numbers and find the average of the 4 numbers.

#include <iostream>

using namespace std;

int main(){

int x[4];

int average = 0;

cout << "Enter four integers: ";

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

cin >> x[i];

average += x[i];

}

cout << "average is: " << average;

}

1. Write a program to read 2 numbers and display the largest of the two. You should use cin to read the two numbers and use if statement to see which number is larger.

#include <iostream>

using namespace std;

int main(){

int x, y;

cout << "Enter two integers: ";

cin >> x;

cin >> y;

if(x > y){

cout << x << " is bigger than the other number.";

}else{

cout << y << " is bigger than the other number.";

}

}

1. Write a program to find the sum of the even numbers starting at 2 ending at 1000. You must use a while loop in your program.

#include <iostream>

using namespace std;

int main(){

int sum;

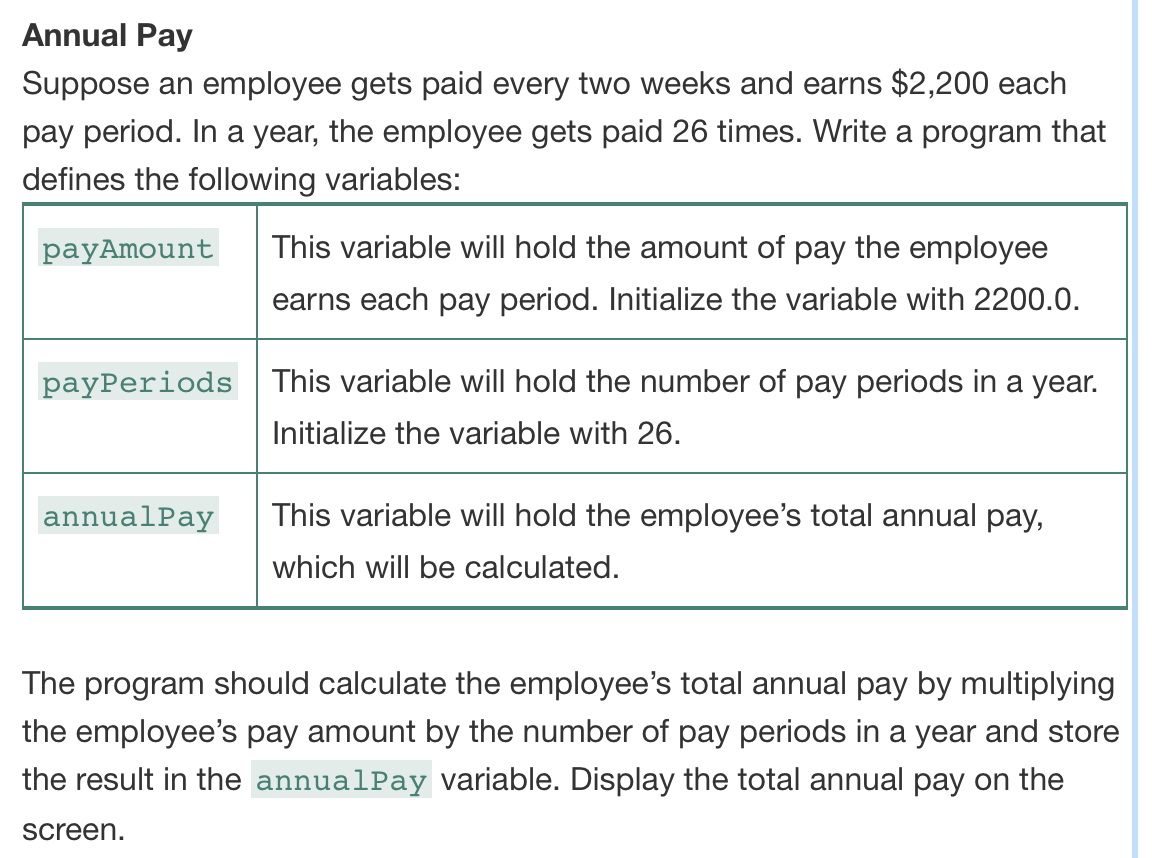
for (int i = 2; i <= 1000; i+=2){

sum += i;

}

cout << "sum: " << sum;

}



#ifndef EMPLOYEEPAY\_H

#define EMPLOYEEPAY\_H

#include <string>

class EmployeePay

{

public:

static double payAmount, payPeriods, annualPay;

double calculateAnnualPay(double, double);

void displayResults();

};

#endif // EMPLOYEEPAY\_H

#include "EmployeePay.h"

#include <iostream>

using namespace std;

double EmployeePay::payAmount = 2200.0;

double EmployeePay::payPeriods = 26;

double EmployeePay::annualPay = 0;

double EmployeePay::calculateAnnualPay(double amount, double periods){

annualPay = amount \* periods;

return annualPay;

}

void EmployeePay::displayResults(){

cout << "Employee's annual pay is: " << annualPay << ".";

}

#include <iostream>

#include "EmployeePay.h"

using namespace std;

int main()

{

EmployeePay eo;

eo.calculateAnnualPay(eo.payAmount, eo.payPeriods);

eo.displayResults();

return 0;

}

