SAKARYA UNIVERSITY

FACULTY OF COMPUTER AND INFORMATICS

SCIENCE DEPARTMENT OF COMPUTER ENGINEERING INTRODUCTION TO PROGRAMMING COURSE 1ST ASSIGNMENT REPORT

Student Number: B231210016

Student Name: Okan

Student Last name: Koca

Lesson Group: C

Homework Subject

Basically, a program that converts grades to letter grades (AA, BA, BB etc.). A program which generates random names, surnames, exam grades, homework grades, final grades according to entered(by user) student number; calculates passing grade according to entered(by user) weights of each lesson and finally converts passing grade to letter grade and gives some statistics of class.

How I have done this assignment:

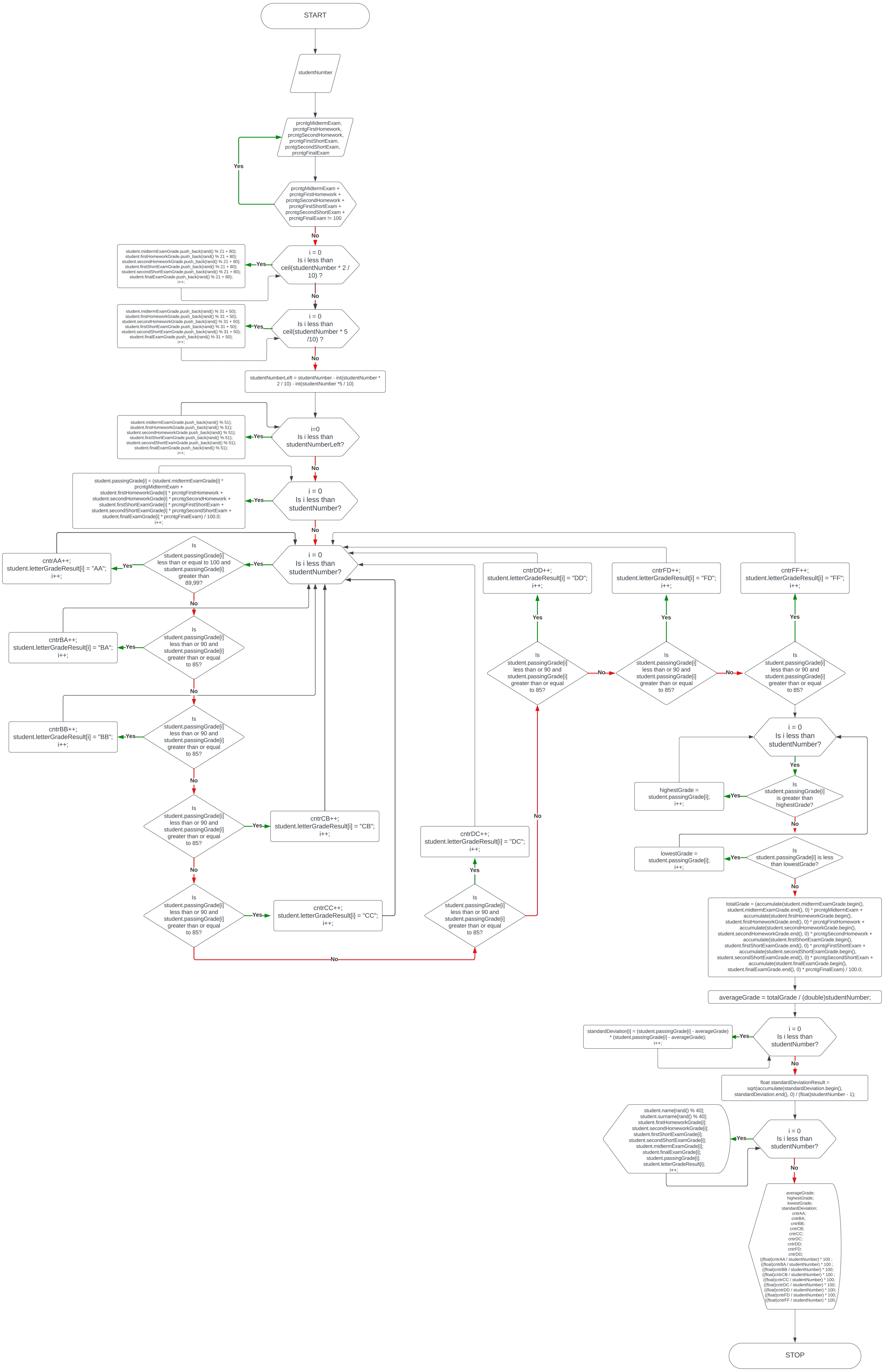
I created a struct named student and two vectors named name and surname which contains forty different names and foty different surnames as members of struct. Thus, we can combine them and create 1600 random full names. Then, I created seven different vectors which contains nothing for each exam, homework, passing grades etc. I intentionally created them empty to add random values. I used srand and rand function to create random values. The program asks for number of students at first. After number of students is entered, program asks for weights of each exams etc. I put this part in a while loop because total weight must be 100. Otherwise it gives a warning message and iterates. After that, program starts to list every students' name, surname, midterm exam grade, first and second homework grades, first and second short exam grades, final grade and passing grade.(I used for loop to list each student and create random values for each student.) According to passing grade it prints letter grades. (I used if statement to compare passing grades and assigning correct letter grade.)

Finally, program gives us some statistics like highest grade, lowest grade, amount of students whose grade is AA, BA etc. percentage of students whose grade is AA, BA etc. (At first, I declared an integer variable named highest grade and initialized with 0, I declared an integer variable named lowest grade and initialized with 100 and used if statement to compare each passing grades with highest grade variable and lowest grade variable.). Before the program close, it prints a simple graph which is made by printing asterisk until asterisk number is equal to number of students whose grade AA, BA etc.

Pseudo Code

```
• START
• SET a struct named student
• SET two string vectors named name and surname which contains 40 different names and surnames as members of struct.
• SET seven integer vectors named midtermExamGrade, firstHomeworkGrade, secondHomeworkGrade, firstShortExamGrade, secondShortExamGrade, finalExamGrade, passingGrade which are empty for now.
• SET two string vectors, one is empty other one will contain letter grades, named letterGrade and letterGradeResult.
• SET float  highestGrade = 0.0, lowestGrade = 100.0
• SET int cntrAA = 0, cntrBA = 0, cntrBB = 0, cntrCB = 0, cntrCC = 0, cntrDC = 0, cntrDD = 0, cntrFD = 0, cntrFF= 0 (these are for counting letter grades.)
• SET int studentNumber, prcntgMidtermExam, prcntgFirstHomework, prcntgSecondHomework, prcntgFirstShortExam, prcntgSecondShortExam, prcntgFinalExam.
• SET a float vector named standardDeviation which is empty for now.
• Use srand function to get random values each time program ran.
• PRINT "Please enter number of students: "
• GET studentNumber
• Use system("CLS") to clear the screen.
• DO
    o PRINT "Enter the weight of midterm exam: "
    o GET prcntgMidtermExam
    o PRINT "Enter the weight of first homework: "
    o GET prcntgFirstHomework
    o PRINT "Enter the weight of second homework: "
    o GET prcntgSecondHomework
    o PRINT "Enter the weight of first short exam: "
    o GET prcntgFirstShortExam
    o PRINT "Enter the weight of second short exam: "
    o GET prcntgSecondShortExam
    o PRINT "Enter the weight of final exam: "
    o GET prcntgFinalExam
    o IF (prcntgMidtermExam+prcntgFirstHomeworK+prcntgSecondHomework+ prcntgFirstShortExam+prcntgSecondShortExam+prcntgFinalExam != 100)
        ▪ PRINT "Invalid value.(Total of weights must be 100.)
    o ENDIF
    o Use system("pause") to provide user to see the warning message.
    o Use system("CLS")   to clear invalid values left from past.
• WHILE (prcntgMidtermExam+prcntgFirstHomeworK+prcntgSecondHomework+ prcntgFirstShortExam+prcntgSecondShortExam+prcntgFinalExam != 100)
• ENDWHILE
• PRINT "Name" "Surname" "Homework" "Homework" "Short Exam" "Short Exam" "Midterm" "Final"
```

```
SET int studentNumberLeft = studentNumber
FOR i = 1 to less or equal to  ceil(studentNumber * 2  / 10) i++
    Use rand function to create random grades between 80-100 and assign each values to midtermExamGrade’s elements.
    Use rand function to create random grades between 80-100 and assign each values to firstHomeworkGrade’s elements
    Use rand function to create random grades between 80-100 and assign each values to secondHomeworkGrade’s elements
    Use rand function to create random grades between 80-100 and assign each values to firstShortExamGrade’s elements
    Use rand function to create random grades between 80-100 and assign each values to secondShortExamGrade’s elements.
    Use rand function to create random grades between 80-100 and assign each values to finalExamGrade’s elements.
ENDFOR
FOR i = 1 to less or equal to  ceil(studentNumber * 5  / 10) i++
    Use rand function to create random grades between 50-80 and assign each values to midtermExamGrade’s elements.
    Use rand function to create random grades between 50-80 and assign each values to firstHomeworkGrade’s elements
    Use rand function to create random grades between 50-80 and assign each values to secondHomeworkGrade’s elements
    Use rand function to create random grades between 50-80 and assign each values to firstShortExamGrade’s elements
    Use rand function to create random grades between 50-80 and assign each values to secondShortExamGrade’s elements.
    Use rand function to create random grades between 50-80 and assign each values to finalExamGrade’s elements.
ENDFOR
studentNumberLeft = studentNumberLeft – int(studentNumber * 2 /10) – int(studentNumber * 5 / 10)
FOR i = 1 to less or equal to  studentNumberLeft i++
    Use rand function to create random grades between 0-50 and assign each values to midtermExamGrade’s elements.
    Use rand funtion to create random grades between 0-50 and assign each values to firstHomeworkGrade’s elements
    Use rand function to create random grades between 0-50  and assign each values to secondHomeworkGrade’s elements
    Use rand function to create random grades between 0-50 and assign each values to firstShortExamGrade’s elements
    Use rand function to create random grades between 0-50 and assign each values to secondShortExamGrade’s elements.
    Use rand function to create random grades between 0-50 and assign each values to finalExamGrade’s elements.
ENDFOR
FOR i = 0 to studentNumber i++
    student.passingGrade[i] = (student.midtermExamGrade[i] * prcntgMidtermExam + student.firstHomeworkGrade[i] * prcntgFirstHomework+ student.secondHomeworkGrade[i] *prcntgSecondHomework + student.firstShortExamGrade[i] * prcntgFirstShortExam + student.secondShortExamGrade[i] * prcntgSecondShortExam +student.finalExamGrade[i] * prcntgFinalExam) / 100.0
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than or equal to 100 and student.passingGrade[i] greater than 89,99
        cntrAA++
        Assign AA grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 90 and student.passingGrade[i] greater than or equal to 85
        cntrBA++
        Assign BA grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than or 85 and student.passingGrade[i] greater than or equal to 80
        cntrBB++
        Assign BB grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 80 and student.passingGrade[i] greater than or equal to 75
        cntrCB++
        Assign CB grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 75 and student.passingGrade[i] greater than or equal to 65
        cntrCC++
        Assign CC grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 65 and student.passingGrade[i] greater than or equal to 58
        cntrDC++
        Assign DC grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 58 and student.passingGrade[i] greater than or equal to 50
        cntrDD++
        Assign DD grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 50 and student.passingGrade[i] greater than or equal to 40
        cntrFD++
        Assign FD grade to student.letterGradeResult[i]
ENDFOR
FOR i= 0 to studentNumber i++
    IF student.passingGrade[i] less than 40
        cntrFF++
        Assign FF grade to student.letterGradeResult[i]
ENDFOR
FOR i = 0 to studentNumber i++
    IF student.passingGrade[i] greater than highestGrade
        highestGrade= student.passingGrade[i]
ENDFOR
FOR i = 0 to studentNumber i++
    IF student.passingGrade[i] less than lowestGrade
        lowestGrade= student.passingGrade[i]
ENDFOR
SET totalGrade is equal to total amount of whole grades multiplied with their weight
SET averageGrade is equal to totalGrade divided by studentNumber
FOR i = 0 to studentNumber  i++
    standardDeviatiom = square of student.passingGrade[i] - averageGrade
ENDFOR
SET float standardDeviationResult = square root of total amount of standardDeviation elements divided by studentNumber – 1
FOR i = 0 to studentNumber i++
    PRINT i+1, random selected name, random selected surname, firstHomeworkGrade[i], secondHomeworkGrade[i], firstShortExamGrade[i], secondShortExamGrade[i], midtermExamGrade[i], finalExamGrade[i], passingGrade[i], letterGradeResult[i]
ENDFOR
Use system(“pause”)
Use system(“CLS”) to clear the screen
PRINT “Average grade: “
PRINT averageGrade
PRINT “Highest Grade:”
PRINT highestGrade
PRINT “Lowest Grade:”
PRINT lowestGrade
PRINT “Standard Deviation: :”
PRINT standardDeviationResult
PRINT “Amount of AA grades:  “
PRINT cntrAA
PRINT “Amount of BA grades:  “
PRINT cntrBA
PRINT “Amount of BB grades:  “
PRINT cntrBB
PRINT “Amount of CB grades:  “
PRINT cntrCB
PRINT “Amount of CC grades:  “
PRINT cntrCC
PRINT “Amount of DC grades:  “
PRINT cntrDC
PRINT “Amount of DD grades:  “
PRINT cntrDD
PRINT “Amount of FD grades:  “
PRINT cntrFD
PRINT “Amount of FF grades:  “
PRINT cntrFF
PRINT “Percentage of AA grades: %”
PRINT (cntrAA / studentNumber) * 100
PRINT “Percentage of BA grades: %”
PRINT (cntrBA / studentNumber) * 100
PRINT “Percentage of BB grades: %”
PRINT (cntrBB / studentNumber) * 100
PRINT “Percentage of CB grades: %”
PRINT (cntrCB / studentNumber) * 100
PRINT “Percentage of CC grades: %”
PRINT (cntrCC / studentNumber) * 100
PRINT “Percentage of DC grades: %”
PRINT (cntrDC / studentNumber) * 100
PRINT “Percentage of DD grades: %”
PRINT (cntrDD / studentNumber) * 100
PRINT “Percentage of FD grades: %”
PRINT (cntrFD / studentNumber) * 100
PRINT “Percentage of FF grades: %”
PRINT (cntrFF / studentNumber) * 100
STOP
```


START
studentNumber
prcntgMidtermExam, prcntgFirstHomework, prcntgSecondHomework, prcntgFirstShortExam, pcntgSecondShortExam, prcntgFinalExam
prcntgMidtermExam + prcntgFirstHomework + prcntgSecondHomework + prcntgFirstShortExam + prcntgSecondShortExam + prcntgFinalExam != 100
Yes
No
i = 0 Is i less than ceil(studentNumber * 2 / 10) ?
Yes
student.midtermExamGrade.push_back(rand() % 21 + 80); student.firstHomeworkGrade.push_back(rand() % 21 + 80); student.secondHomeworkGrade.push_back(rand() % 21 + 80); student.firstShortExamGrade.push_back(rand() % 21 + 80); student.secondShortExamGrade.push_back(rand() % 21 + 80); student.finalExamGrade.push_back(rand() % 21 + 80); i++;
No
i = 0 Is i less than ceil(studentNumber * 5 /10) ?
Yes
student.midtermExamGrade.push_back(rand() % 31 + 50); student.firstHomeworkGrade.push_back(rand() % 31 + 50); student.secondHomeworkGrade.push_back(rand() % 31 + 50); student.firstShortExamGrade.push_back(rand() % 31 + 50); student.secondShortExamGrade.push_back(rand() % 31 + 50); student.finalExamGrade.push_back(rand() % 31 + 50); i++;
No
studentNumberLeft = studentNumber - int(studentNumber * 2 / 10) - int(studentNumber *5 / 10)
i=0 Is i less than studentNumberLeft?
Yes
student.midtermExamGrade.push_back(rand() % 51); student.firstHomeworkGrade.push_back(rand() % 51); student.secondHomeworkGrade.push_back(rand() % 51); student.firstShortExamGrade.push_back(rand() % 51); student.secondShortExamGrade.push_back(rand() % 51); student.finalExamGrade.push_back(rand() % 51); i++;
No
i = 0 Is i less than studentNumber?
Yes
student.passingGrade[i] = (student.midtermExamGrade[i] * prcntgMidtermExam + student.firstHomeworkGrade[i] * prcntgFirstHomework + student.secondHomeworkGrade[i] * prcntgSecondHomework + student.firstShortExamGrade[i] * prcntgFirstShortExam + student.secondShortExamGrade[i] * prcntgSecondShortExam + student.finalExamGrade[i] * prcntgFinalExam) / 100.0; i++;
No
i = 0 Is i less than studentNumber?
Yes
Is student.passingGrade[i] less than or equal to 100 and student.passingGrade[i] greater than 89,99?
Yes
cntrAA++; student.letterGradeResult[i] = "AA"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrBA++; student.letterGradeResult[i] = "BA"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrBB++; student.letterGradeResult[i] = "BB"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrCB++; student.letterGradeResult[i] = "CB"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrCC++; student.letterGradeResult[i] = "CC"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrDC++; student.letterGradeResult[i] = "DC"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrDD++; student.letterGradeResult[i] = "DD"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrFD++; student.letterGradeResult[i] = "FD"; i++;
No
Is student.passingGrade[i] less than or 90 and student.passingGrade[i] greater than or equal to 85?
Yes
cntrFF++; student.letterGradeResult[i] = "FF"; i++;
i = 0 Is i less than studentNumber?
Yes
Is student.passingGrade[i] is greater than highestGrade?
Yes
highestGrade = student.passingGrade[i]; i++;
No
Is student.passingGrade[i] is less than lowestGrade?
Yes
lowestGrade = student.passingGrade[i]; i++;
No
totalGrade = (accumulate(student.midtermExamGrade.begin(), student.midtermExamGrade.end(), 0) * prcntgMidtermExam + accumulate(student.firstHomeworkGrade.begin(), student.firstHomeworkGrade.end(), 0) * prcntgFirstHomework + accumulate(student.secondHomeworkGrade.begin(), student.secondHomeworkGrade.end(), 0) * prcntgSecondHomework + accumulate(student.firstShortExamGrade.begin(), student.firstShortExamGrade.end(), 0) * prcntgFirstShortExam + accumulate(student.secondShortExamGrade.begin(), student.secondShortExamGrade.end(), 0) * prcntgSecondShortExam + accumulate(student.finalExamGrade.begin(), student.finalExamGrade.end(), 0) * prcntgFinalExam) / 100.0;
averageGrade = totalGrade / (double)studentNumber;
i = 0 Is i less than studentNumber?
Yes
standardDeviation[i] = (student.passingGrade[i] - averageGrade) * (student.passingGrade[i] - averageGrade); i++;
No
float standardDeviationResult = sqrt(accumulate(standardDeviation.begin(), standardDeviation.end(), 0) / (float)studentNumber - 1);
i = 0 Is i less than studentNumber?
Yes
student.name[rand() % 40]; student.surname[rand() % 40]; student.firstHomeworkGrade[i]; student.secondHomeworkGrade[i]; student.firstShortExamGrade[i]; student.secondShortExamGrade[i]; student.midtermExamGrade[i]; student.finalExamGrade[i]; student.passingGrade[i]; student.letterGradeResult[i]; i++;
No
averageGrade; highestGrade; lowestGrade; standardDeviation; cntrAA; cntrBA; cntrBB; cntrCB; cntrCC; cntrDC; cntrDD; cntrFD; cntrDD; ((float)cntrAA / studentNumber) * 100 ; ((float)cntrBA / studentNumber) * 100 ; ((float)cntrBB / studentNumber) * 100; ((float)cntrCB / studentNumber) * 100 ; ((float)cntrCC / studentNumber) * 100; ((float)cntrDC / studentNumber) * 100; ((float)cntrDD / studentNumber) * 100; ((float)cntrFD / studentNumber) * 100; ((float)cntrFF / studentNumber) * 100;
STOP