

EXERCISE SET 1**PROGRAMMING WORKSHOP I, ACADEMIC YEAR 2019-2020****Deadline: 24/11/2019, 23:59****Read before you start**

Read the ENTIRE pronunciation carefully and "draw" your program on paper. If you don't know the concepts of mean, and standard deviation, see [here](#).

Decide what variables you will need, what names you will give them, whether they need constants and if so for what quantities, what control structures you will use for each function, how and at what point in the program you will do the required calculations. Each time you complete a stage, confirm that your program is working properly before moving on.

If you need clarification or have problems, send a message to the e-class discussion area. Warning: posting code to eclass is not allowed.

This task can be done in groups of up to 2 people. You don't have to be in a group with the same person you are in the lab with. You can discuss the exercises with your fellow students but you are not allowed to exchange code in any way.

If you work in a group, then before starting work log into Autolab, select hw1 and build a group via Group Options. Don't leave this process for the last minute!

Start early! Planning is always MUCH more time consuming than you expect.

Late submissions of exercises are not accepted.

Your exercises will be graded on the following (in no particular order):

- Correctness of calculations
- Effective use of appropriate structures, variables, constants, etc.
- General program formatting (alignment, variable and constant names, etc.)
- Compliance with specifications
- Effective comments

Use of goto and use of global variables is strictly prohibited.

You can assume that keyboard input will always be given in the correct format, for example if a number is requested a string will not be given.

Automatic scoring of multiple choice tests

In this exercise you will write a program that checks the answers of a number of students on a multiple-choice test and produces statistics about the students' performance.

General information

The test contains NUM_QUESTIONS questions. All questions have the same number of options, which correspond to the characters 'A'-'Z'. The first choice is always 'A', followed by the rest in the same order as they appear in the English alphabet. The special character '-' means that a question has not been answered.

The program manages NUM_STUDENTS students taking the test.

For each student, you will need to store the number of questions they answered correctly, the number of questions they answered incorrectly, and their grade. Use a two-dimensional array of double-precision floating-point numbers, each row of which will contain the above information for a student.

You should decide if and what other tables are needed to store data.

Files

We give you two files:

- hw1.h contains the definition of four constants that you will use in the program
your:
 - NUM_STUDENTS: number of students (default value 50)
 - NUM_QUESTIONS: number of questions (default value 20)
 - MIN_CHOICE: first possible choice (default value 'A')
 - MAX_CHOICE: last possible choice (default value 'Z')
- hw1.c is the file in which you will write your program. It contains the directive `#include "hw1.h"` through which the contents of hw1.h are included in hw1.c so that you can use the above constants in your program and the directive `#include <math.h>` which is necessary for the use of mathematical functions.

You can, as part of your program's sanity checks, change the values of the constants defined in hw1.h. No further changes or addition of additional code to this file are allowed. You are not allowed to redefine these constants in hw1.c.

Output messages

In all messages that end with a colon, there is a space after it.

Where we tell you to print the delimiter string, print `"\n##\n"`

Math functions

To compile math functions you should add `-lm` (hyphen el em) to the end of the gcc command.

Program mode

Data entry

Print a newline character and the message "Enter number of choices (1-X): where X is the maximum number of choices (integer) and read from the keyboard the actual number of choices (integer). As long as this is not in the range 1 to X, prints and enters the count again. For example, if MIN_CHOICE is 'A' and MAX_CHOICE is 'Z' then the message will be Enter number of choices (1-26):

Based on the value given by the user, calculate the maximum actual option. For example, if the user gives 5, then the maximum actual choice will be 'E'.

Print a newline character and the message "Enter answer key: " (once only) and read from the keyboard the correct choice for one question at a time (with a space between them, which is ignored). If given input that is not within the range of options as specified above, then print a newline character and the message "Error. Enter value YZ: ", where Y is the minimum possible option and Z is the maximum actual option, and reread the option. Printing the error message and reading is repeated until a valid value is given. Continuing the example above, the message should be: Error.

Enter AE:

Print a newline character and the message "Enter wrong answer penalty: read from the keyboard the point " and penalty for wrong answers, which is a non-positive floating point number. If the given value is positive, then print a newline character and the message " Error. Enter non-positive value: and reread the answer. Printing the error message and reading is repeated until a valid value is given.

For each student, print a newline character and the message "Enter student choices: " (once only) and read the student's choices (answers) for each question one by one (with a space between them which is ignored). If a choice is given that is neither in the correct choice range nor +, then print a newline character and the message "Error. Enter valid choice:

" and re-read the choice for that question.

Calculations

For each student, calculate the number of correct and number of incorrect choices and store these values in the appropriate locations in the student data table.

For each student, calculate his grade obtained from the formula:

$$(0, 10) \frac{(\text{number of correct answers}) + (\text{number of wrong answers}) \cdot (\text{penalty})}{(\text{number of questions})} \max$$

and save it to the student data table.

Calculate the grade point average of the students.

Calculate the standard deviation of the scores given by the formula:

$$\sqrt{\frac{\sum_{i=0}^{n-1} (\text{grade}[i] - \text{average})^2}{n}}$$

where n is the number of students. To calculate the square root of an expression x and store it in a variable y , you write $y = \text{sqrt}(x)$. To calculate the square of an expression x and store it in a variable y , you write $y = \text{pow}(x, 2)$.

Student grades are always in the range 0-10. Count the number of students whose grade falls in each of the intervals [0, 1), [1, 2), ..., [8, 9), [9, 10] and store these values in an integer array 10, in which cell 0 corresponds to degrees belonging to the interval [0, 1), cell 1 to degrees belonging to the interval [1, 2) and so on. Think of a clever way to populate the table – not with sequential if statements.

You'll find the math library's floor function useful: Calling floor(x) for a real number x returns the largest integer value not greater than x. For example, floor(3.1) is 3.0. Similarly, floor(3) is 3.0. Because the return type of floor is double, you will need to convert the return value to the corresponding integer with an appropriate typecast.

This table will be used later to print the grade histogram.

Functions menu

After successfully reading and inputting data, repeatedly print function menus and manage each function appropriately.

To be precise, print a newline character, the following menu:

```
[M/m] Print mean
[S/s] Print standard deviation
[G/g] Print grade range
[I/i] Print student info in range
[H/h] Draw histogram
[Q/q] Quit
```

and newline character.

Read the user's selection from the keyboard. Valid options are the characters shown in the menu in brackets. If an invalid option is given, print a newline character and the message "Error. Invalid option." and repeat menu printing and selection reading.

On Q or q, the program terminates.

In the M or m option, print a newline character, the average with two decimal places, and the delimiter string.

In the S or s option, print a newline character, the standard deviation to two decimal places, and the delimiter string.

In the G or g option, print a newline character and the message "Enter grade range: and read from the keyboard two integers, either low, high, separated by dashes. If the integers read are not in the interval [0, 10] or the first is not less than the second, printing the message and reading values is repeated. When valid values are given, count the number of students whose grade is in the interval [low, high) or especially for the case where high is equal to 10, in the range [low, high]. Print newline character, the count you calculated, and the delimiter string.

In the I or i option, print a newline character and the message "Enter index range: and read from the keyboard two integers separated by a dash. The integers represent a range of student array positions. If the first is not less than the second or the range is outside the bounds of the array, printing the message and reading values is repeated. When valid values are given, print the following information based on the student data in the first through second integer positions of the array:

- For each student:

- newline character,
- position in the table (index) with a width of 3 characters and leading zeros
- colon
- number of correct answers with a width of 4 characters, without decimal places,
- comma,
- number of incorrect answers with a width of 4 characters, without decimal places,
- comma,
- number of unanswered questions with a width of 4 characters, without decimal places,
- comma,
- degree with two decimal places and a total width of 6 characters,
- At the end, the delimiter string.

In option H or h, print a histogram that "graphically" shows how many students scored in each interval [0, 1), [1, 2), ..., [8, 9), [9, 10] as calculated earlier. In fact, for each interval, the program prints:

- left bracket
- the lower limit of the range corresponding to this position with a width of 2 characters,
- comma,
- the upper limit of the range corresponding to this position with a width of 2 characters
- right parenthesis , except in the case of 10, where a right bracket is printed,
- colon
- a gap
- as many asterisks (*) as there are X's of students who scored in this range, where
 - Xs=units if the total number of students who took the test is at most 100
 - X's=tens if the total number of students who took the test is greater than 100

After printing the histogram, the program prints the delimiter string.

Guidelines for submitting work to Autolab

1. Add in comments at the beginning of the hw1.c file the full names and AEMs of its members group. Please write comments in English characters ONLY.
2. Make a directory named hw1submit and copy hw1.c into it
3. Make a text file named team.txt and add to it the full names and IDs of the team members, even if the team consists of one person.
4. Right-click the hw1submit directory and select Compress here as tar.gz. I will create a file named hw1submit.tar.gz.
5. Log in to Autolab and select hw1.
 - If you are a group of two and have not already done so, create a group via Group Options.
6. Submit hw1submit.tar.gz .