Assignment 7: pointers

In this assignment you need to create a program which process students' data in a university course. In order to do that you need to implement these functions:

int find_failures(int grades[], char *names[], int students_num, char *failed[], double *average, int *maximal)
the function gets as input 2 arrays ("names" and "grades") where the first one holds the names of the students and the second one holds their grades. For each 0 <= j < students_num, grades[j] holds the grade for the student names[j]. The output of the function is:

1. Names of failed students (those who got grade which is less than 60). The names will be returned in array named "failed" which contains pointers to names of failed students in "names" array. You can assume you have enough cells in "failed" to hold all the pointers to the relevant cells in "names".
2. The average of the course (will be saved inside the average argument).
3. The maximal grade in the course (will be saved inside the maximal argument).
4. The returned value of the function will be the number of failed students (the length of "failed" array).

int validate_data(int grades[], char *names[], int size)
The function will get as input 2 arrays – grades and student names – and the size of the arrays (they have the same size). The function will validate that the entered data is valid - grades are between 0 and 100 and names are made of English characters. In case that a name or a grade is invalid it has to be deleted from the 2 arrays – both the name and the relevant grade. Then, the other arrays cells have to move one index lower (empty cells can be only in the end of the arrays). For example, if grade[j] is invalid, it will be deleted and names[j] also will be deleted. Then, grades[j+1] and names[j+1] will be moved to grades[j] and names[j], grades[j+2] and names[j+2] will be moved to grades[j+1] and names[j+1] until the end of the arrays. To avoid duplication the

unused cells in the end of the array (if there are any) will be filled with '\0' for names and with -1 for grades. The returned value is the new size of the arrays (after deletion).

Write a program which get grades and names for 6 students. The data will be entered sequentially in one line – first name and grade for first student, then name and grade for second student etc. The program will validate the input data (using validate_data()) and then - using find_failures() – the program will calculate the requested values (average, maximal value, failed array). In main(), the program will print to the user course average, maximal grade, number of failures and names of failed students.

Remark: For reading the students names from the user you need, first, to create 2 dimensions array of characters (for holding the names in memory) and only then you should use *names array to point to this names.

Running example (program in **red**, user in **blue**):

Please input students' data:

Yossi 59 Chedva 94 Rivka 46 Ran 99 Amir 43 Zohar 79

Average grade is 70

Maximal grade is 99

3 students failed: Yossi, Rivka, Amir