Computer Systems Organization
Assignment 1: C Programming
Due: September 21, 11:55 PM

1 Code of Conduct

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your work with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi (see https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/).

2 Questions

Question 1
Write a program that:
1. Creates a structure called "university_detail" containing the university's name.
2. Creates another structure called "student_detail" that contains the student's id, name, gpa, and a university_detail struct.
3. Creates an array of structures of student_details of size 3.
4. Creates a function that prints the details of all students in the array.

Test your program by creating 3 students and filling all its required data from the user.

Sample Input/Output
Enter the name of the first student
Bob
Enter the id of the first student
2016
Enter the gpa of the first student
3.2
Enter the name of the university
LAU
Enter the name of the second student

Tod
Enter the id of the second student
2017
Enter the gpa of the second student
3.86
Enter the name of the university
AUB
Enter the name of the third student
ALice
Enter the id of the third student
20
Enter the gpa of the third student
3.5
Enter the name of the university
NYU
Student Bob with ID 2016 and GPA 3.20 is studying at LAU
Student Tod with ID 2017 and GPA 3.86 is studying at AUB
Student ALice with ID 20 and GPA 3.50 is studying at NYU

Question 2
Implement a C program that prints all the numbers that appear in an array once and
only once. The ç] ^Á¡ -Àó@ Áarray • @ ˘ |åÀå^Áẙ cẠ̊} åÁó@ sizeÁ -Àó@ Áṡ¦ẩ Á @ ˘ |åÀå^ 1€.
The values • ̨ ¦^åÁẙ the array should be taken from the ˘•^¦Áó@[˘ * @Á̂^¦{ ̆} ̨æ|È
Examples:
  - If the input is 1 2 2 3 H3 GI F Í then the output should be [1,2,3 Ê Ễ].
  - If the input is 7 6 6 F Ï F I I G8 then the output should be [7,6,F Ễ Ê Ĝ B].

Question 3
Implement a C program that removes the last occurrence of a character in a string. Whether the
character appears only once or multiple times, the last occurrence of that character should be
removed from the string.
Examples:
  - If the input is "12233" then the output should be "23".
  - If the input is "122331" then the output should be "123".
  - If the input is "123" then the output should be "".
  - If the input is "12233177719" then the output should be "123177".

Question 4
Implement a C program that reads a string from the user and prints it repeatedly from the end
and adding one character at a time.
For example, "abcde" will be printed as such:

        e
        de
        cde
        bcde
        abcde

3. Grading

- The whole assignment is worth 40 points, 10 points per question. The overall grade will be then converted over 10.

- The correct version of each program is worth 6.5 points. You lose 1 point for each missed test case. That is, for each time we try a test input and the output of your program is incorrect. The test cases will not be provided in the assignment. It is up to you to write/guess a set of possible test cases and try them yourself on your program before submission to make sure that your program is behaving as expected. Your goal here should be to cover a reasonable number of input cases based on the problem statement (this is known as software testing and is an important practice in software engineering).

- Efficiency of the solution is worth 1.5 points. A solution is efficient if it has a lower complexity, is faster in execution or requires less memory than a naive solution. There is no particular specification of what the naïve solution is (as in real life), it is up to you to think about possible improvements of your solution to make it more efficient. A general guideline is the following: your solution is considered inefficient if improving its efficiency (complexity, execution time and memory footprint) is trivial (requires one hour or two of work).

- Fully commented code with clear explanation is worth 2 points.

4 Submission

Please submit one zip file containing all your **C** programs and screenshots of your outputs all included in one **PDF** file on Brightspace (https://brightspace.nyu.edu/). Submissions via email are not accepted.

Good luck! ☺