COP3275 – Spring 2019

Instructor: Roozbeh Ketabi

Programming Assignment 4

1- You have to upload your codes in both Canvas and the Judge system at https://cop3275.cise.ufl.edu

2- For guide on how to access the Judge, visit Judge Access Page under Pages section in Canvas.

3- The judge will run test cases against your code and assign it a grade. You can have multiple submission for the same problem but you have to choose which one is the final (we will consider your final submission for grading). Canvas is only used for record keeping.

4- When upload on canvas, zip your programs in a zipfile with your ufid as name (nothing more, just 8 digit ufid, for instance 12345678.zip). Name your files (inside the zipped archive) p1.c, p2.c, and so on.

5- For all the problems input is read from standard input (i.e. read using scanf) and must be written to standard output (i.e. printf).

6- **Don't print extra stuff. Since the assignments are automatically graded, if the output doesn't match the expected output, you will not get the points. For instance:**
If the problem is to read a number and return its square, the expected output is a number (i.e. "printf("%d",i*i)"). If you print using something like "printf("square is %d",i*i);" you will not receive any points.

7- The assignment is due on 11:59 pm Sunday March 31st 2019. There is a 20% penalty for late submission of one day. No submission is accepted beyond that time.

Problem 1: Reverse words (2 pts.)

For this problem you are given a string (char array) representing a sentence. Reverse the words (separated by space) in the sentence (the last word in original sentence appears first and so on). Assume punctuations are part of the word (i.e. “how are you?” becomes “you? are how”. In other words, any thing between spaces or start or end of string are considered words). Print the new sentence. Don’t change the letters cases.

Examples:

Input	**Output**
Hi, how are you?	you? are how Hi,
aa bb cc dd	dd cc bb aa
Keep up the good work!	work! good the up Keep

Hints:

- You can tokenize the input on space and store tokens in an array of strings (array of char arrays). See strtok function for this. This function will modify the given input.
- Traverse the array in reverse order and concatenate the words together with a space in between. You can use strcat for this purpose.

- ❖ There is another tricky way to solve this problem. Write a function that takes in a string and two indices indicating the start and end of a substring of that string, and reverses the letters in that substring (i.e. if you pass in 0 as start and length-1 as end, it will reverse the letters of the whole string, e.g. “hi there” becomes “ereht ih”).
- ❖ Think about how you can use this function to solve the problem (we will talk about it in class).
- ❖ You MUST write your own function to reverse a substring. You **cannot** use strrev as it is NOT part of the standard C.

Problem 2: Student Management System (9 pts.)

In this problem you are to implement a system that keeps track of students, courses and enrollments. Your system should be able to have enrollment and unenrollment capability and it should be able to answer the queries described below:

Syntax	**What it means?**
enroll student_firstname student_lastname course_name	Enroll the student in that course. If already enrolled do nothing.
unenroll student_firstname student_lastname course_name	Drop that course for that student. If not enrolled do nothing.
grade student_firstname student_lastname course_name grade	Grade that student for that course! If not enrolled already do nothing.
ave course_name	Ave grade of the class (ignore ungraded). If no one is graded, print 0.
gpa student_firstname student_lastname	Compute the gpa of the student in their current courses (assume every course has same number of credits). If not graded print 0.
count course_name	Total number of enrollments in the given class.
topstudent course_name	Find the top student of the class. If no one is graded, print “N/A”. If more than one student are top students (same maximum grade), print them separated by comma and space.
findmutual student_firstname student_lastname	Name of the students who take two or more classes together with the given student. Print first name and last names (with a space between them) separated by a comma (and space). If no one is found print “N/A”.
listcourses student_fname student_lastname	List courses that the student is enrolled in. Separate each with a comma and a space (e.g. “programming in c, java fundamentals”). Print “N/A” if not enrolled in any.
findfirstnames student_lastname	Print all the students’ **first names** that share the same last name (comma space separated). Print “N/A” if no one matches the last name.

First, the number of students (N) and courses (M) is given in the first line (space separated). Then the name of the students is given in the next N following lines. First name and last name is given separated by space in each line. Then the name of the courses follows in M lines (a course name may contain spaces so the whole line will be the course’s name).

Then comes a series of queries with the syntax defined above. For each query, print the result in a newline (except enroll, unenroll and grade). Queries can be either upper or lower case. One way of handling this, is to change everything to lowercase first and have selection statements on lowercase string literals. **In all cases**, if there are multiple results (course or student names), print them separated by **comma-space** (e.g. john doe, jane doe, john smith) in a **newline**. Everything should be printed in lower case. All the floating point numbers must be printed to **exactly** two decimal places. Exit the program if command “**quit**” is given in input.

Example:

Input	Output
5 3 john doe jane doe john smith jane smith rooz beh programming using c analysis of algorithms general math 1 findfirstnames smith enroll john doe programming using c enroll jane doe programming using c grade john doe programming using c 90 grade jane doe programming using c 100 ave programming using c unenroll john doe programming using c enroll john smith programming using c enroll john smith analysis of algorithms enroll jane doe analysis of algorithms grade jane doe analysis of algorithms 85 grade john smith programming using c 100 gpa jane doe findmutual john smith listcourses john smith topstudent programming using c count analysis of algorithms topstudent general math 1 topstudent analysis of algorithms quit	john, jane 95.00 92.50 jane doe programming using c, analysis of algorithms jane doe, john smith 2 N/A jane doe