COL100

Lab 13

Note: You need to upload all your program C files as a single zip file on moodle by 8:00pm 12/11/2017.

Q1. Write a C function to read the following space separated information (from the standard input) about students of IIT-Delhi and return it using a suitably defined structure:

First Name (maximum 64 chars) Date of birth (in format ddmmyyyy) Height in millimetres (integer) weight in kg (float).

Date of birth should be another structure with three integers corresponding to day, month and year.

Input format: First line contains a number representing the number of records in the file.

The next lines contain the data of one student per line with each entry separated using a comma. You may read the input using scanf("%s", ...) or any other library function. Test your function by extracting the first 4 records from the sample input provided on moodle (input.txt) and creating a suitable input file. While running your program, redirect its input from the file that you created using the "<" linux command.

Example of the commands that you might execute on Linux.

From the input.txt provided, create a file called example.txt containing the first 4 records.

head -5 input.txt > example.txt

Now edit your example.txt and replace the number in the first line by 4.

Now run your program by redirecting the example.txt to its standard input.

my_program < example.txt

Q2. Modify your structure so that it may be used as a node in a linked list of records of students. Use your function and the malloc() function to create a linked list of records to be read from standard input. Write a recursive function to print the linked list starting from the last node to the first node. Test your program on the sample input provided by redirecting the file input.txt to the standard input of the program.

my_program < input.txt

Q3. Write a function that takes two date of birth structures as input and returns -1, 0, 1 if the first parameter is less than, equal to or greater than the second parameter respectively.

Using your function, write a recursive mergeSort program to sort the linked list of student records in increasing order of their age (do not use arrays, use linked list only). Print all the records starting from the first to the last. Run the program and redirect your output to the file output_mergeSort.txt

my_program < input.txt > output_mergeSort.txt

Q4. Write a function that takes two student record structures by reference and swaps all their contents except their next pointers. Use your functions to implement bubble sort algorithm to sort the linked list (do not use arrays). Run the program and redirect your output to the file output_bubbleSort.txt

my_program < input.txt > output_bubbleSort.txt

Compare the two outputs by using the diff command. The two outputs should be identical and diff should not print anything.

diff output_mergeSort.txt output_bubbleSort.txt

Q5. Use your functions to implement quick sort algorithm to sort the linked list (do not use arrays). Run the program and redirect your output to the file output_quickSort.txt

my_program < input.txt > output_quickSort.txt

Compare the two outputs by using the diff command. The two outputs should be identical and diff should not print anything.

diff output_mergeSort.txt output_quickSort.txt

Q6. Using strcmp() function in the standard c library <string.h>, change all the three sorting programs written by you to sort and print the records in ascending lexicographic ordering of names. If you have done it right, in each of your programs the change should be less than one line each.

Compare the output of all the three programs. They should be identical.