COL100 Lab 13

I semester 2016-17

Week 14, 2016

Objective

To be able to write C programs involving structures and pointers.

Instructions

1. There are multiple ways to achieve a task in C. Please follow the method that the questions asks you to do.

Examples

Functions

Here is an example how to create a simple structure, containing the name, entry number and COL100 marks of a student:

```
struct student {
   char name[40];
   char entryNumber[11];
   float mark;
};
```

Now suppose that we create an array called as *studentRecords*, each element of which is of type *student*. This can be done as follows:

```
student studentRecords[] = {
   { "Abhishek", "2016CSZ10301", 77.0 } ,
   { "Atul", "2016CSZ10302", 83.3 } ,
   { "Peter", "2016CSZ10456", 53.5 } ,
   ...
   { "Rekha", "2016CSZ10777", 61.3 }
};
```

Now, to access the marks of Atul, we can use $studentRecords[1].mark$.

Assume that we have multiple entries in alphabetical order, of the name, entry number and marks of the students. For example,
Abhishek, 2016CSZ10301, 77.0
Atul, 2016CSZ10302, 83.3
Peter, 2016CSZ10456, 53.5
...

Programs

1. Assume that there are 100 entries, containing the name, entry number and marks of students. Your task is to create the array, *studentRecords*, and store the entries in it in the same order as they appear in the input. Calculate and print the mean of marks obtained by all the students, and the maximum marks obtained.

   We have provided a list of 100 entries at the location `http://www.cse.iitd.ernet.in/~hameedah/col100/records.txt`. You can read them into your code using redirection as: "records.txt < a.out". No need to use file I/O.

2. Now pass this array of structures to a function called as *sortRecords*() using pointers to structures, and sort the structures in order of increasing marks. Print the modified array of structures.

3. The same task can be achieved in a different way. Instead of creating an array of structures, we can create an array of pointers to structures. We can declare a pointer to the structure *student* using :

   ```
   struct student *ptr;
   ```

   Now we can dynamically allocate memory for this pointer, based on the number of records we want to store. We can then successively store each record by incrementing the pointer each time. Do this, and again calculate and print the mean of marks obtained by all the students, and the maximum marks obtained.

   Next, sort the arrays in increasing order of marks, by passing a pointer to the first record to a function called *sortRecords*2(). Print the records.

4. Next, assume that we have a list of friendship relations, (x,y) meaning that x and y are friends of each other. Now, add an element to the structure *student* called *numFriends*, which stores the count of friends for each student. Calculate and print the maximum, minimum and mean number of friends the students have.

5. Now, also find out the number of friends of friends that each students has, and store it in another element in the structure *numFrFriends*.

6. Suppose that you are given two students, find a path that they can follow to get introduced to each other.

Optional Problems

1. Also find out the number of friends of friends that each students has, and store it in another element in the structure *numFrFriends*.

2. Suppose that you are given two students, find a path that they can follow to get introduced to each other.

Useful Commands in Linux

1. Open terminal: Ctrl + Alt + T

2. Terminate current Linux command: Ctrl + C

3. Make a new directory: mkdir *dirname*

4. Copy: cp *src dest*

5. Rename: mv *originalname newname*

6. Delete: rm *filename*

7. Change working directory: cd *path*

8. List contents of a folder: ls

9. List contents of a folder including hidden files: ls -a

10. Print current directory: pwd

Points to Remember

1. To set proxy: Open an internet browser and set the Automatic proxy configuration url to http://www.cc.iitd.ernet.in/cgi-bin/proxy.btech (or proxy.dual if you are a Dual Degree student).
   (For Firefox, open Options > Advanced > Network Tab > (Connection) Settings > Choose "Automatic proxy configuration" and set the URL)

Optional : Use vim editor

1. Open a file: vim filename.txt
2. Insert in a file: i (insert mode) (Use Esc to come out of the insert mode)
3. Navigation: arrow keys
4. Undo u
5. Redo Ctrl+R
6. Saving a file :w
7. Closing a file without saving :q!
8. Saving and closing a file :wq
9. Deleting a line dd
10. Copying a line yy
11. Pasting a line p