CS 2213 Advanced Programming Recitation

Due date: check BB
!!!! NO LATE HOMEWORK WILL BE ACCEPTED !!!
Total 5 points

(Library – Recursion)

In this assignment, you are asked to first implement `ListPermutations` that we studied in class. Then using the `ListPermutations` example as a starting point, write another function `ListSubsets` that generates all possible subsets of a given set, where the set is again represented by a string of letters. For example, if you call the function `ListSubsets("ABC");` your function should produce the following output:

Subsets:

```
{ABC}
{AB}
{AC}
{A}
{BC}
{B}
{C}
{}
```

Like permutations, the subset problem has a recursive formulation. If you represent a set of characters using a string that either contains or does not contain a given letter, you can calculate all possible subsets by (1) including the first character in the subset and concatenating it onto the front of all subsets of the remaining N–1 characters and then (2) displaying the subset of the remaining N-1 without this character. (Note: This is the programming exercise 6 in Chapter 5).

To separate the implementation of the above two functions from the driver/client program, **define** and **implement** an interface called `permutation.h` that exports two functions:

```
void ListPermutations(string str); // and
void ListSubsets(string str);
```

The implementation of these functions and the implementation of any other subsidiary functions or utility functions that you may need will be in `permutation.c`.

Finally you will implement a client/driver program (e.g., `driver.c`) that asks user to enter a string `str` where all characters are different, then call `ListPermutations(str)` and `ListSubsets(str)`. You need to make sure user enters different characters; otherwise, you will keep asking him to enter a string with different characters.

As always, make sure you release (free) the dynamically allocated memories if you allocate any in your programs. So, before submitting your program, run it with `valgrind` to see if there is any memory leakage… Also if you need to debug your program, compile your programs with –g option and then run it with `gdb` and/or `ddd`.

/* Don't forget to include comments about the problem, yourself and each major step in your program! */

What to return: !!!! NO LATE HOMEWORK WILL BE ACCEPTED !!!

1. Create a directory, say LASTNAME_ch05_recursion, and do all your work under that directory.
2. You will implement a simple library (`permutation.h` and `permutation.c`) and use this library along with other libraries in your driver/client program, say `driver.c`.
3. To easily compile the library and driver program, you must have a `Makefile` and use "`make`" to compile your code.
4. After compiling, run your program a few times and save the output (using script) into `output.txt` file.

So you will have around 6-7 files in your LASTNAME_ch05_recursion directory.

5. Go to parent directory of LASTNAME_ch05_recursion, and use

```
> tar -cf LASTNAME_ch05_recur.tar  LASTNAME_ch05_recursion
```

This will create a new file called LASTNAME_ch05_recur.tar and it contains all of your files. So just submit this .tar file.

6. Go to WebCT (BB), and just submit LASTNAME_ch05_recur.tar as **attachment** before the deadline. DO NOT submit other .h or .c files individually.

You must submit your work using Blackboard Learn and respect the following rules:

1) All assignments must be submitted as either a zip or tar archive file unless it is a single pdf file.
2) Assignments must include all source code.
3) Assignments must include an output.txt file which demonstrates the final test output run by the student.
4) If your assignment does not run/compile, the output.txt file should include an explanation of what was accomplished, what the error message was that prevented the student from finishing the assignment and what the student BELIEVES to be the underlying cause of the error.
