CSCI 689 **Computer Assignment 11** **Fall 2017**

(10 points)

Displaying Permutations

For this computer assignment, you are to write a C++ program to generate and display all permutations of the characters of a string on stdout. In addition, you are also to generate and display all permutations of the characters of all substrings of the given string.

In your program, in addition to the main ( ) routine, implement the following subroutines:

- set < string, cmp > genPerm (const string& str): generates all permutations of the string str and inserts them on the returned set using the following recursive strategy, which is different than the strategy presented in the textbook:

  (a) Removes the first character from str and store it in the variable c.
  (b) Generates the set containing all permutations of the remaining characters.
  (c) Forms a new set by inserting c in every possible position in each of those permutations.

- set < string, cmp > genSub (const string& str): generates all substrings of the string str and for each of those substrings generates all permutations by calling the routine genPerm ( ) and inserts them on the returned set.

- void print (const set < string, cmp >& s, const unsigned& sz): prints the contents of the set s on stdout by allocating sz spaces for each printed item. For the output, do not include the empty string when you're displaying all permutations, and to eliminate the empty string, you can start with the iterator ++s.cbegin ( ) for the set s. The maximum number of elements can be printed on a single line is LSIZE = 10, whose definition is included in the header file prog11.h.

Programming Notes:

(1) The prototypes of all above subroutines are given in the header file prog11.h, so in your source file prog11.cc, you need to insert the following line at the top of your source file: #include "/home/cs689/progs/17f/p11/prog11.h".

(2) The elements in a declared set will be sorted by the sorting criteria cmp, which is the second template argument for the set, and its implementation (as a class) is also given in the header file prog11.h.

(3) This is an interactive program. In other words, the main ( ) routine gets the input string from the stdin; however, for a final test, your program should get the input string from the input file prog11.d, which is in the same directory with the header file prog11.h.

(4) The main ( ) routine calls the routine genSub ( ) to get all permutations of the characters of all substrings of the input string, and it calls the routine genPerm ( )

to get all permutations of the characters of the input string and stores them in the final set to display, and it also obtains the size of the input string and passes that value as the second argument to the print ( ) routine.

(5) To compile your source file and link the generated object file with the system library routines, execute: Make N=11. For a final test of your program, execute: Make execute N=11. This will test your program and generate the output file prog11.out. The correct output file, prog11.out, which is in the same directory with the header file prog11.h.

(6) When your program is ready, mail your source file to your TA by executing: mail_prog.689 prog11.cc.

(7) <u>If your implementation of your program does not use the strategy specified in this handout but uses the strategy outlined in the textbook, no credit will be given to your program</u>.