CSCI 340 **Computer Assignment 8** **Fall 2017**
(10 points)

Heaps and Heapsort

For this computer assignment, you are to write a C++ program to sort items in several input files, using the heapsort technique. For each input file, your program first reads the items from the input file and builds a heap structure for these items. Then, it retrieves these items from the heap structure in order and prints them out on stdout. Full path names of the input files, certain constant definitions, and the prototypes of the function templates are included in the header file prog8.h, in directory: ~cs340/progs/17f/p8.

Implement the following function templates in your program:

- template < class T > void get_list (vector < T >& v, const char* path): It opens an input file for reading, then reads the items from the file and inserts them in a vector. Finally, it closes the input file. The first argument v to this function is the vector and the second argument path is the full path name of the input file.

- template < class T, class P > void construct_heap (vector < T >& v, P pred): It constructs a heap structure from the items of vector v and uses the predicate pred to compare the items when building the heap. It calls the function make_heap ( ) to construct the heap and the function sort_heap ( ) to sort the items in the heap using the predicate pred. Both functions are from the STL.

The definition of the print_list class is included in the header file prog8.h. Implement its member functions (as templates) in your program:

- template < class T > print_list < T > :: print_list (const unsigned& s, const unsigned& w, const unsigned& l, const unsigned& c): The constructor of the print_list class, where s is the heap size, w is the minimum number of chars written in printout, l is the maximum number of items printed in a single line, and c is used as a counter with the default value 0 that can be used to insert the newline characters in printout.

- template < class T > void print_list < T > :: operator ( ) (const T& x): It can be used to print the item x of a heap on stdout. For proper printout, insert the following statements at the beginning of this function:

```
cout.width ( wid ); cout.precision ( 2 );
cout << fixed << showpoint;
```

The binary predicates greater < > ( ) and less < > ( ) (in the STL) take two arguments and return a Boolean value to the calling routine. The call greater < T > (x, y) takes the arguments x and y as inputs, and if x is greater than y, it returns true; otherwise, it returns false; and the call less < T > (x, y) returns true if x is less than y; otherwise, it returns false. To build a heap, the driver program uses these two predicates.

The main ( ) routine for your program is given in the source file prog8–main.cc, in the same directory with prog8.h. Copy this routine into your source file prog8.cc, and then add implementations of the subroutines defined above. This routine acts as a driver for your heap structure, where several vector objects are defined.

To compile and link the driver program with the system library routines, execute: Make N=8. To test your program, execute: Make execute N=8. After you are done with the program, you don't need its object and executable files any more. To delete them, execute: Make clean.

The correct output for this program is in file prog8.out, which is in the same directory with prog8.h.

When your program is ready, submit its source file to your TA, executing: mail_prog.340 prog8.cc.