CSCI 340 **Computer Assignment 2** **Fall 2017**
(10 points)

Search Algorithms

For this computer assignment, you are to write and implement a C++ program to implement two search algorithms (a *linear search* and a *binary search*) on randomly generated integers stored in vectors.

Put the implementations of your subroutines (described below) in your source file sub2.cc and their prototypes in your header file prog2.h.

- void Vectors (vector < int >& v1, vector < int >& v2, int s1, int s2) : Fills the elements of vectors v1 of size ARR_SIZE = 200 and v2 of size TEST_ARR_SIZE = 100 with random numbers, generated by two sets of pseudo–random numbers with the seed values s1 and s2, where s1 (defined as SEED1 = 1) is for v1and s2 (defined as SEED2 = 3) is for v2. To initiate a random number generator (RNG) with the seed value seed, execute the system function srand (seed) (only once), and to generate a random integer in the range [LOW = 1, HIGH = 1000], execute: rand ( ) % (HIGH – LOW + 1) + LOW.

- bool linearSearch (const vector < int >& v, int x) : A linear search algorithm, where x is the searched item in vector v. It simply starts searching for x from the beginning of vector v to the end, but it stops searching when there is a match. If the search is successful, it returns true; otherwise, it returns false. To implement this routine, simply call the find ( ) function in the STL.

- bool binarySearch (const vector < int >& v, int x) : A binary search algorithm, where x is the searched item in vector v. If the search is successful, it returns true; otherwise, it returns false. To implement this routine, simply call the binary_search ( ) function in the STL.

- int search (const vector < int >& v1, const vector < int >& v2, bool (*p) (const vector < int >&, int)) : A generic search algorithm – takes a pointer to the search routine p ( ), and then it calls p ( ) for each element of vector v2 in vector v1. It computes the total number of successful searches and returns that value to the main ( ) routine as an input argument to the print routine printStat ( ), which is used to print out the final statistics for a search algorithm.

- void sortVector (vector < int >& v) : A sort algorithm to sort the elements of vector v in ascending order. To implement this routine, simply call the sort ( ) function in the STL.

- void printVector (const vector < int >& v) : Prints the contents of vector v on stdout, up to NO_ITEMS = 16 items on a single line except perhaps the last line. The sorted numbers need to be properly aligned on the output. For each printed number, allocate ITEM_W = 4 spaces for each item.

- void printStat (int totalSucCnt, int vectorSz) : Prints the percent of successful searches as right-aligned, floating-point numbers on stdout, where totalSucCnt is the total number of successful comparisons and vectorSz is the size of the test vector.

Programming Notes:

- Include your header file prog2.h, by inserting the statement: #include "prog2.h" at the top of your source file sub2.cc.

- Make a link to the source file of the driver program prog2.cc, by executing the statement: ln –s ~cs340/progs/17f/p2/prog2.cc.

- To compile and link your program with the system library routines, execute: Make N=2. To test your program, execute: Make execute N=2. This command executes the driver program and displays the output of the program as well as any error messages both on the terminal screen and in prog2.out. You can find all related files for this computer assignment in directory: ~cs340/progs/17f/p2. After you are done with the program, you don't need its object and executable files any more. To delete them, execute: Make clean.

- You can find the correct output of this program in file prog2.out in directory: ~cs340/progs/17f/p2.

- You are not allowed to use any I/O functions from the C library, such as scanf or printf. Instead, use the I/O functions from the C++ library, such as cin or cout.

- When your program is ready, submit your source and header files to your TA by executing: mail_prog.340 sub2.cc prog2.h.