Assignment 4

Due. Nov. 9 (Wednesday), 17:30.

Programming style (10 marks).

1. (6 marks) Chapter 6, Programming 5, p. 226.
   Update the permutation algorithm from the text to generate the correct list of permutations even if the string contains repeated letters. For example, if you call `ListPermutations` on the string "AABB", your program should not generate as many permutations as it does for the string "ABCD" because some of the strings generated by the standard algorithm would be indistinguishable from others. Your program should instead generate the following six:

   ```
   AABB
   ABAB
   ABBA
   BAAB
   BABA
   BBAA
   ```

   Write a new implementation of `ListPermutations` that works correctly even if the string contains duplicated letters. In writing this implementation, you should not merely keep a list of permutations that have already be encountered and avoid generating duplicates. Instead, you should think carefully about the recursive structure of the problem and find a way to avoid generating the duplicates in the first place. You may assume the string contains only upper case letters from A to Z.

2. (10 marks) Chapter 6, Programming 15, p. 233
   Recursive decomposition can also be used to draw a stylized representation of a tree. The tree begins as a simple trunk indicated by a straight vertical line. The trunk may branch at the top to form two lines veer off at an angle, as shown in the figure on page 234. These branches may themselves split to form new branches, which split to form new ones, and so on. If the decision to branch is made randomly at each step of the process, the tree will eventually become unsymmetrical and will end up looking a little more like trees in nature, as illustrated by the diagram on page 234.

   If you think about this process recursively, however, you can see that all trees constructed in this way consist of a trunk, optionally topped by two trees that veer off at an angle. If the probability of branching is a function of the length of the current branch, the process will eventually terminate as the branches get progressively shorter.

   Write a program `drawtree.cpp` that uses this recursive strategy and the graphic library to draw a stylized line drawing of a tree.

3. (12 marks) Chapter 7, Programming 7, p. 274.
   In chess, a knight moves in an L-shaped pattern: two squares in one direction horizontally or vertically, and then one square at right angles to that motion. For example, as shown in the diagram on page 274, the knight in the upper right side of the diagram can move to any of the eight squares marked with a black cross. The mobility of a knight decreases near the edge of the board, as illustrated by the bottom knight, which can reach only the three squares marked by white crosses.

   It turns out that a knight can visit all 64 squares on a chessboard without ever moving to the same square twice. A path of the knight that moves through all the squares without repeating a square is

called a *knight's tour*. One such tour is shown in the diagram on page 275, in which the numbers in the squares indicate the order in which they were visited.

Write a program that uses backtracking recursion to find a knight's tour.