CISC-365*
2009
Lab # 1
Week of September 14

It is often useful to generate all permutations of a set of objects (as we will discuss in class). Your task in this lab is to implement a particular non-recursive algorithm for generating the permutations of the set of integers {1, 2, ..., n} where n is a positive integer provided as input to your program.

In this algorithm, each integer in the set is assigned a "facing direction" which can be either "left" or "right". Initially, the integers are in ascending order (i.e. the first permutation is 1, 2, ... n) and all the integers are facing "left".

An integer is "mobile" if it is larger than the neighbour it is facing.

The algorithm can be expressed in just a few lines:

```
Output the initial permutation
While there is at least one mobile integer in the set {
        let k be the largest mobile integer
        swap k with the neighbour it is facing
        for each integer m that is larger than k, reverse the facing direction of m
        output the new permutation
} // while
```

Implement this algorithm in the language of your choice. The integer n should be entered by the user, and the full set of permutations of {1, 2, ..., n} should be displayed on the screen.

For further consideration: This is not formally part of the lab assignment, but it is an important question. What is the computational complexity of this algorithm?