EE231002 Introduction to Programming

Lab06. Permutations

Due: Oct. 29, 2022

Given a set of N distinct elements, there are $N!$ possible permutations to arrange these elements. For example, given the set {A, B, C} they can be arranged in the following 6 permutations:

$$\begin{array}{ccc} A & B & C \\ A & C & B \\ B & A & C \\ B & C & A \\ C & A & B \\ C & B & A \end{array}$$

Your assignment is to write a program to generate all possible permutations given the set of N letters starting from A using Pandita algorithm. The output should follow the format below.

Assuming the array `A[N]` stores a permutation, the following algorithm by 14th century Narayana Pandita of India produces the next lexicographic permutation:

1. Find the largest index `j` such that `A[j] < A[j + 1]`.
   If no such index exists, the permutation is the last permutation and the program should terminate.
2. Find the largest index `k` such that `A[j] < A[k]`.
3. Swap `A[j]` with `A[k]`.
4. Reverse the sequence from `A[j + 1]` up to and including the last element `A[N - 1]`.

If the array `A` is initialized to $\{A, B, \cdots\}$, then repeatedly applying Pandita's algorithm it would generate all possible permutations. Please implement Pandita's algorithm such that all possible permutations are generated in the following format:

Example output (N=4):

```
$ a.out
permutation #1: A B C D
permutation #2: A B D C
permutation #3: A C B D
permutation #4: A C D B
permutation #5: A D B C
permutation #6: A D C B
permutation #7: B A C D
permutation #8: B A D C
permutation #9: B C A D
permutation #10: B C D A
permutation #11: B D A C
permutation #12: B D C A
permutation #13: C A B D
permutation #14: C A D B
permutation #15: C B A D
permutation #16: C B D A
permutation #17: C D A B
permutation #18: C D B A
permutation #19: D A B C
permutation #20: D A C B
permutation #21: D B A C
permutation #22: D B C A
permutation #23: D C A B
permutation #24: D C B A
  Total number of permutations is 24
```

Please note that your loop of finding all permutations should be terminated using the step 1 of the Pandita algorithm.

Notes.

1. Create a directory **`lab06`** and use it as the working directory.

2. Name your program source file as **`lab06.c`**.

3. The first few lines of your program should be comments as the following.

   ```
   // EE231002 Lab06. Permutations
   // ID, Name
   // Date
   ```

4. The number of elements, `N`, should be defined as a macro as following:

   ```
   #define N 7
   ```

5. After finishing editing your source file, you can execute the following command to compile it,

   ```
   $ gcc lab06.c
   ```

   If no compilation errors, the executable file, **`a.out`**, should be generated, and you can execute it by typing

   ```
   $ ./a.out
   ```

6. After you finish verifying your program, you can submit your source code by

   ```
   $ ~ee2310/bin/submit lab06 lab06.c
   ```

   If you see a ”submitted” message, then you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following command:

   ```
   $ ~ee2310/bin/subrec lab06
   ```

   It will show the last few submission records.