EE231002 Introduction to Programming

Lab05. Permutations

Due: Oct. 25, 2017

Given a set of N distinct elements, there are N! possible permutations to arrange these elements. For example, given the set $\{1, 2, 3\}$ they can be arranged in the following 6 permutations:

Your assignment is to write a program to generate all possible permutations given the set of N integers from 1 to N in reverse order. The output should follow the format below.

Assuming the array A[N] stores a permutation, the following algorithm generates the next permutation in reverse lexicographic order. The algorithm is based on the one invented by Narayan Pandia of India in 14th century.

- 1. Find the largest index j such that A[j]>A[j+1]. If no such index exists, the permutation is the last permutation.
- 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 $\{N, N-1, \dots, 1\}$, then repeatedly applying the algorithm it would generate all possible permutations. Please implement the algorithm such that all possible permutations are generated in the following format:

Example output (N=4):

permutation #1: 4 3 2 1 permutation #2: 4 3 1 2 permutation #3: 4 2 3 1 permutation #4: 4 2 1 3

permutation	#5: 4 1 3 2
permutation	#6: 4 1 2 3
permutation	#7: 3 4 2 1
permutation	#8: 3 4 1 2
permutation	#9: 3 2 4 1
permutation	#10: 3 2 1 4
permutation	#11: 3 1 4 2
permutation	#12: 3 1 2 4
permutation	#13: 2 4 3 1
permutation	#14: 2 4 1 3
permutation	#15: 2 3 4 1
permutation	#16: 2 3 1 4
permutation	#17: 2 1 4 3
permutation	#18: 2 1 3 4
permutation	#19: 1 4 3 2
permutation	#20: 1 4 2 3
permutation	#21: 1 3 4 2
permutation	#22: 1 3 2 4
permutation	#23: 1 2 4 3
permutation	#24: 1 2 3 4
Total num	ber of permutations is 24

Notes.

- 1. Create a directory **lab05** and use it as the working directory.
- 2. Name your program source file as lab05.c.
- 3. The first few lines of your program should be comments as the following.

```
/* EE231002 Lab05. 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 lab05.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

$\sim ee231002/bin/submit lab05 lab05.c$

If you see a "submitted successfully" 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:

$\sim ee231002/bin/subrec lab05$

It will show the last few submission records.

