lab05

```
$ gcc lab05.c
$ ./a.out
permutation #1: 1 2 3 4 5 6 7_
permutation #1: 1 2 3 4 5 6 7
permutation #2: 1 2 3 4 5 7 6_
permutation #2: 1 2 3 4 5 7 6
permutation #3: 1 2 3 4 6 5 7_
permutation #3: 1 2 3 4 6 5 7
permutation #4: 1 2 3 4 6 7 5_
permutation #4: 1 2 3 4 6 7 5
...
...
permutation #5037: 7 6 5 4 2 1 3_
permutation #5037: 7 6 5 4 2 1 3
permutation #5038: 7 6 5 4 2 3 1_
permutation #5038: 7 6 5 4 2 3 1
permutation #5039: 7 6 5 4 3 1 2_
permutation #5039: 7 6 5 4 3 1 2
permutation #5040: 7 6 5 4 3 2 1_
permutation #5040: 7 6 5 4 3 2 1
  Total number of permutations is 5040

CPU time: 0.00892581 sec
score: 91
o. [Output] Program output is incorrect
o. [Format] Program format can be improved
o. [Efficiency] can be improved.
```

lab05.c

```c
// EE2310 lab05 Permutations
// 109061217 林峻霆
// Date: 2020/10/26

#include <stdio.h>

#define N 7

int main(void)
{
    int j;                                  // a parameter in loop
    int k;                                  // a parameter in loop
    int i = 0;                              // a parameter in loop
    int tmp;                                // a variable for swapping
    int A[N];                               // array to store data
    int total = 1;                          // number of permutation

    for (; i < N; i++) {                    // input data in array
        A[i] = i + 1;
    }
    printf("permutation #%d: ", total);
    for(i = 0; i < N; i++) {                // the first permutation
    for (i = 0; i < N; i++) {                // the first permutation
        printf("%d ", A[i]);
    }
    printf("\n");

    i = N - 2;                              // change i to N - 2
    while (i >= 0) {                        // initialize a loop
        if (A[i] < A[i + 1]) {              // compare the data
            j = i + 1;                      // change j to i + 1
            total = total + 1;
            for (; j < N ; j++) {           // swap two data
            for ( ; j < N; j++) {           // swap two data
                if (A[i] > A[j]) {
                    tmp = A[i];
                    A[i] = A [j - 1];
                    A[i] = A[j - 1];
                    A[j - 1] = tmp;
                    j = N;                  // jump out the loop
```

```
38                  }
39                  else if (j == N - 1) {            // the final-element case
40                      tmp = A[i];
41                      A[i] = A[j];
42                      A[j] = tmp;
43                  }
44              }
45              j = i + 1;                            // change j to i + 1
46              for (k = 0; j + k < N - 1 - k; k++) { // reverse part of the array
47                  tmp = A[j + k];
48                  A[j + k] = A[N - 1 - k];
49                  A[N - 1 - k] = tmp;
50              }
51              printf("permutation #%d: ", total);
52              for (k = 0; k < N; k++) {             // print the permutation
53                  printf("%d ", A[k]);
54              }
55              printf("\n");
56              i = N - 2;                            // change i back to N - 2
57          }
58          else
59              i = i - 1;                            // see next element in array
60      }
61      printf("  Total number of permutations");     // print the total amount
62      printf(" is %d\n", total);
63
64      return 0;                                     // end the program
65  }
```