

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
    Total number of permutations is 5040

CPU time: 0.00899355 sec
score: 79
o. [Output] Program output is incorrect
o. [Format] Program format can be improved
o. [Coding] lab05.c spelling errors: store(1)
o. [Efficiency] can be improved.
o. [Program] should terminate by step 1 checking.
o. [Step 2] should not be embedded in step 1.
```

lab05.c

```
1 // EE231002 Lab05 Permutations
2 // 109061158, 簡佳吟
3 // Date: 2020.10.26
4 // Date: 2020/10/26
5
6 #include <stdio.h>
7 #define N 7
8
9
10 int main(void)
11 {
12
13     int i, j, k;           // index i, j, k
14     int foundj, foundk;    // indication for finding j and k
15     int a[N];             // array
16     int temp;              // store number temporarily
17     int serial = 1;        // serial number of solutions
18
19
20     for (i = 0; i < N; i++) {
21         a[i] = i + 1;      // assign number from 1 to N
22     }
23
24
25     for (i = 0; i < N; i++) {
26         printf("permutation #%d: ", serial++);
27         for (i = 0; i < N; i++) {
28             printf("%d ", a[i]);
29         }
30         printf("\n");      // prompt output numbers
31         foundj = 0;        // assign 0 to foundj
32         foundk = 0;        // assign 0 to foundk
33
34         for (j = N - 2; j >= 0 && !foundj; j--) {
35             if (a[j] < a[j + 1]) {                // find j
36                 for (k = N - 1; k > 0 && !foundk; k--) {
37                     if (a[j] < a[k]) {            // find k
38                         temp = a[j];
39                         a[j] = a[k];
```

```

40         a[k] = temp;                                // swap a[i] and a[k]
41         foundk = 1;                                  // break
42         for (i = 1; i <= (N - j - 1) / 2; i++) {
43             temp = a[j + i];
44             a[j + i] = a[N - i];
45             a[N - i] = temp;                          // reverse the number
46                                                         // from j + 1 to N - 1
47
48         }
49
50     }
51 }
52     foundj = 1;                                       // break
53     foundj = 1;                                       // break
54 }
55 }
56
57
58 }
59
60     printf("    Total number of permutations is %d\n", serial - 1 ); // prompt
61     printf("    Total number of permutations is %d\n", serial - 1); // prompt
62
63     return 0;                                         // done and return
64
65 }
66
67
68
69
70
71
72
73
74
75
76
77
78

```

79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101