

Introduction to Computers and Programming LAB-10 2014/12/03

- ✧ Your output must be in our sample output format.
 - ✧ In **Problem 1~4**, please wrap each of your code inside `main(){ }` with `while(1){ }`
1. Please finish the `lab_10_1.c` program in which finds the maximum value and the second large value from the **positive integers** array. You need to get the value by using pointers.
Note: you can ONLY modify the function: **`void max_sec(int[], int, int*, int*);`**

```
Input
N: 5
Array: 1 2 3 4 5

max = 5
sec = 4
請按任意鍵繼續 . . .

Input
N: 4
Array: 2 7 9 1

max = 9
sec = 7
請按任意鍵繼續 . . .
```

2. Finish the `lab_10_2.c` program in which you will enter the array elements until your input is -1. The program will reverse the array (you don't need to reverse -1). Please write your code in the function "`reverse(int* start, int* end)`".

```
Before reverse:
2 3 4 -1
After reverse:
4 3 2

Before reverse:
6 5 4 3 -1
After reverse:
3 4 5 6

Before reverse:
```

3. Please finish the `lab_10_3.c` program in which you will enter one sequence and its size. The program will print all the permutation combinations of this sequence. If there have duplicate combinations, you can still print them out.

```

Input
N: 3
sequences: 1 2 3

1
2
1 2
3
1 3
2 3
1 2 3
請按任意鍵繼續 . . .
Input
N: 3
sequences: 1 1 2

1
1
1 1
2
1 2
1 2
1 1 2
請按任意鍵繼續 . . .

```

4. Write a program of Hanoi Tower with **recursive function**.

The Tower of Hanoi is a well-known mathematical puzzle. It consists of 3 rods which is called 'A', 'B' and 'C', respectively, and n number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.



The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.

3. No disk may be placed on top of a smaller disk.

4. You have to move all disks from rod A to rod C.

(Hint: before moving the n disks from rod A to rod C,

there are three steps:

1. Move n-1 disks from rod A to rod B.
2. Move the nth disk from rod A to rod C.
3. Move n-1 disks from rod B to rod C.)

You will have to output each step and the number of total runs.

```
Enter the height : 2
1: Move the 1st disk from A to B
2: Move the 2nd disk from A to C
3: Move the 1st disk from B to C
Totally move 3 times
```

```
Enter the height : 3
1: Move the 1st disk from A to C
2: Move the 2nd disk from A to B
3: Move the 1st disk from C to B
4: Move the 3rd disk from A to C
5: Move the 1st disk from B to A
6: Move the 2nd disk from B to C
7: Move the 1st disk from A to C
Totally move 7 times
```