Recursive Problems

นิสิตส่ง pseudocode ให้อาจารย์ approve ก่อนค่อยเริ่มทำ

`int *a` และ `char *a` ในภาษา C หมายถึง `pointer (memory address)`

1. Find min, given an array.

```
int FindMin(int *start, int *end) {
   // return the minimum element in the array
   // start and end are pointers to the first and the last array elements
}
```

2. Sort from min to max, given an array.

```
void Sort(int *start, int *end) {
   // return nothing, but sort the array
   // start and end are pointers to the first and the last array elements
}
```

3. Binary search, given a sorted array and a key (found/not found).

```
int BinarySearch(int key, int *start, int *end) {
   // if key found, return the address of the key
   // if key not found, return -1
   // start and end are pointers to the first and the last array elements
}
```

4. Find the GCD of two numbers.
   Hint: using only addition (+) and subtraction (-) is sufficient, no needs for modulation (%) and division (÷).

```
int GCD(int x, int y) {
   // return the greatest common divisor of x and y
}
```

5. Find the LCM of two numbers.
   Hint: using only addition (+) and subtraction (-) is sufficient, no needs for modulation (%) and division (÷).

```
int LCM(int x, int y) {
   // return the least common multiple of x and y
}
```

6. Is a prime number? (false = 0, true = 1)

```
bool isPrime(int n, int i = 2)
{
    // Base cases
    if (n <= 2)
        return (n == 2) ? true : false;
    if (n % i == 0)
        return false;
    if (i * i > n)
        return true;

    // Check for next divisor
    return isPrime(n, i + 1);
}
```

7. Binary to decimal. For example, the decimal of 13_{10} is 1101_{10}, the decimal of 20_{10} is 10100_{10}
   Hint: use 16-bit Stack simulator for large numbers.

```
int B2D(int x, int d = 0) {
   if (x == 0) return 0;
   return ((x % 2) * pow(10, d)) + B2D(x / 2, d + 1);
}
```

8. Find the sum of digits. For example, the digit sum of 12_{10} is 2_{10}, and the digit sum of 4563_{10} is 4_{10}.
   Hint: use 16-bit Stack simulator for large numbers.

```
int DigitSum(int x, … … …) {
   // return the sum of decimal digits of x
   // for instance, the digit sum of 12₁₀ is 2₁₀, the digit sum of 4563₁₀ is 4₁₀
}
```

9. Is a palindrome string? (false = 0, true = 1) For example, anna, deified, deleveled.

```
bool IsPalindromeString(char *start, char *end) {
   // if it is a palindrome string, return true
   // otherwise return false
}
```

10. Is a palindrome number? (false = 0, true = 1) For example, 303_{10} or 12321_{10} or 745547_{10}.

```
int IsPalindromeNumber(int x, … … …) {
   // if x is a palindrome number, return true
   // otherwise return false
}
```

11. Find X^Y.

```
int Power(int x, int y) {
   // return x^y
}
```

12. Tower of Hanoi. Stack CPU: calculate the number of moves, LC3 processor: print all moves.

```
int Hanoi(int n) {
   // Hanoi(1) = 1
   // Hanoi(n) = (2 * Hanoi(n - 1)) + 1
}

void Hanoi(int n, int src, int des, int tmp) {
   // Hanoi(2, 1, 2, 3)
   //    1 to 3
   //    1 to 2
   //    3 to 2
}
```

13. Find the n^{th} Fibonacci number.

```
int Fib(int n) {
   // return the n^th Fibonacci number
}
```

14. Reverse a string. For example, the string “abcde” is reversed to “edcba.”

```
void Reverse(char *start, char *end) {
   // return nothing, but reverse the string
}
```

15. Find the factorial of n. For example, 5! = 120.

```
int Fac(int n) {
   // return n!
}
```