

Data Structure Homework 2

繳交期限： 2020/10/20 17:00 前

補交期限(7 折)： 2020/10/27 17:00 前

手寫題：

(題目中 write an algorithm 請寫 pseudo code)

3. Consider the following algorithm:

```
algorithm fun3 (x, y)
1  if (x > y)
    1  return -1
2  elseif (x equal y)
    1  return 1
3  else
    1  return (x * fun3 (x + 1, y))
4  end if
end fun3
```

What would be returned if fun3 is called as

- a. fun3 (10,4)?
- b. fun3 (4,3)?
- c. fun3 (4,7)?
- d. fun3 (0,0)?

6. Ackerman's number, used in mathematical logic, can be calculated using the formula shown in Figure 2-17. Write a recursive algorithm that calculates Ackerman's number. Verify your algorithm by using it to manually calculate the following test cases: Ackerman(2, 3), Ackerman(2, 5), Ackerman(0, 3), and Ackerman(3, 0).

$$\text{Ackerman}(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ \text{Ackerman}(m - 1, 1) & \text{if } n = 0 \text{ and } m > 0 \\ \text{Ackerman}(m - 1, \text{Ackerman}(m, n - 1)) & \text{otherwise} \end{cases}$$

8. Write a recursive algorithm that converts a string of numerals to an integer. For example, "43567" will be converted to 43567.
14. Write a recursive algorithm that finds all occurrences of a substring in a string.

程式題 (為第 22 題) :

22. Write the iterative version of the Fibonacci series algorithm using the hints given in Project 21. Note that step c in Project 21 will be different because factorial uses two recursive calls in the last statement.

-- (參考 Project 21)

21. If a recursion call is the last executable statement in the algorithm, called tail recursion, it can easily be removed using iteration. Tail recursion is so named because the return point of each call is at the end of the algorithm. Thus, there are no executable statements to be executed after each call. To change a tail recursion to an iteration, we use the following steps:
 - a. Use a variable to replace the procedure call.
 - b. Use a loop with the limit condition as the base case (or its complement).
 - c. Enclose all executable statements inside the loop.
 - d. Change the recursive call to an appropriate assignment statement.
 - e. Use appropriate statements to reassign values to parameters.
 - f. Return the value of the variable defined in step a.

Write the iterative version of the recursion factorial algorithm (Algorithm 2-2) and test it by printing the value of `factorial(5)` and `factorial(20)`.

-- (參考 Algorithm 2-2)

ALGORITHM 2-2 Recursive Factorial

```
Algorithm recursiveFactorial (n)
Calculates factorial of a number using recursion.
Pre    n is the number being raised factorially
Post   n! is returned
1 if (n equals 0)
1 return 1
2 else
1 return (n * recursiveFactorial (n - 1))
3 end if
end recursiveFactorial
```

-- 程式要求：

測值代入任意 n

Ex:

程式 output：

Enter n：

n

f(n)

Enter n: -1

Error!!!

Enter n: 0

0	1
---	---

Enter n: 10

0	1
1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34
9	55
10	89