VG101 Lab Worksheet

Lab 4

Instructor: Dr. Yifei ZHU
TA: Hangrui CAO
TA: Qinhang WU
TA: Muchen XU

Code Quality Requirement

- Ensure proper indentation
- Make naming meaningful
- Include necessary comments
- Split the code over functions
- Test the code as much as you can

Category	Exercise title	Tag
basic syntax exercise	Start-up	recursion
basic syntax exercise	1 to 1 Recursion	recursion
basic syntax exercise	1 to n Recursion	recursion
basic syntax exercise	Struct	struct, fileI/O
practical exercise	Division	recursion
practical exercise	Eight Queens	recursion, search
practical exercise	Image Deblurring	documentation, plot, image-processing

- In-lab quiz will test your understanding of some of these problems, get prepared.
- Remember to submit your code on time, otherwise your attendance score will be deducted.
- Coding is not that difficult. Think before programming to see whether there is an easier implementation. Make your life easier.

Basic Syntax Exercise

Recursion

0. Predict the output of the following script. Explain how function call itself by comparing your prediction and the actual output:

```
disp(func(3));
function res=func(x)
    fprintf('Entering func(%d)\n',x);
    if x<=1    % Boundary
        res=1;
    else       % Recursion Stage
        res=func(x-1)+func(x-2);
    end
    fprintf('Exiting func(%d) with result=%d\n',x,res);
end
```

1. Explain how recursion works by considering these questions:
    - What is the order of result we (or the computer) want to know?
    - What is the order of result actually being calculated out?
    - How recursion resembles mathematical induction?
        - Boundary and recursion stage are both important!

1 to 1 Recursion

2. Use recursion to find the factorial of n.
3. Use recursion to rewrite Simple Sum in Lab 2. (Calculate $\sum_{i=1}^{n} i^3$)
4. Use recursion to find n^m.
5. Use recursion to find the greatest common divisor of a and b by Euclidean algorithm. Example 3.3.2 in the website clearly illustrated this algorithm.
6. Prove that such "1 to 1" recursion can also be written by iteration.

Program skeleton you may use:

```
disp(fac(6));
disp(simpleSum(5));
disp(power(3,4));
disp(mygcd(72,48));

% Do not modify code above

function res=fac(n)
    % TODO: calculate fac(n) with fac(n-1)
end

function res=simpleSum(n)
    % TODO: calculate simpleSum(n) with simpleSum(n-1)
end

function res=power(n,m)
    % TODO: calculate power(n,m) with power(n,m-1)
end

function res=mygcd(n,m)
    % TODO: calculate mygcd(n,m) with mygcd(m,n%m)
end
```

Expected output is

```
   720

   225

    81

    24
```

1 to n Recursion

7. Use recursion to find the m^{th} Fibonacci number.
8. Use recursion to output all binary numbers with m digits.

9. Use recursion to output all ternary numbers with m digits.
10. Use recursion to output all permutation of $1 \ldots m$.
11. Consider why not all "1 to n" recursion can be (easily) written by iteration.

Program skeleton you may use:

```
disp(fib(12))
mybin('',3);
disp(' ');
myter('',3);
myperm([],3);

% Do not modify code above

function res=fib(m)
    % TODO: calculate fib(m) with fib(m-1) and fib(m-2)
end
function mybin(s,m)
    if length(s)==m
        disp(s);
    else
        % TODO: augment s with possible entries ('0' and '1'),
        %       and pass it to mybin() again.
    end
end
function myter(s,m)
    % TODO: imitate the previous function,
    %       output all ternary numbers with m digits.
end
function myperm(s,m)
    % TODO: output all permutation of  1..m.
end
```

Expected output is like

```
   233

000
001
...
111

000
001
002
010
...
221
222
     1     2     3

     1     3     2

     2     1     3

     2     3     1

```

```
    3     1     2

    3     2     1

```

Struct and File IO

After taking GV010 Mid1, students' scores for three different questions are stored in a text file. The first line represents the number of students. The next few lines represent the score of three problems for each student respectively:

"input.txt"

```
4
100 100 100
200 100 400
50 50 50
0 0 0
```

Write a function, read a given `input.txt`, store certain data into a struct, and output all of the information of the student that obtains the highest total score to `output.txt`.

"output.txt"

```
700 200 100 400
```

You may use the code skeleton below:

```
getBest("input.txt","output.txt");

function getBest(inputFileName,outputFileName)
    % TODO: read scores from file called "inputFileName";
    %       find the student with maximal total score;
    %       then output the total score as well as subscores
    %       to the outputFileName;
end
```

Note that `s = setfield(s,'field',value)` is equivalent to `s.field = value`. So if `s` does not contain the specified field, the `setfield` function creates the field and assigns the specified value.

Now as the semester is proceeding, we would like to update the score structure. Use `setfield` to setup a new score structure including semester, subject, student and scores.

Practical Exercises

Division

Divide number n into k numbers $a_1 \cdots a_k$, such that

$\sum_{i=1}^{k} a_i = n.$

Find how many different methods are there. Two division a, b are considered same if they can be rearranged to be the same.

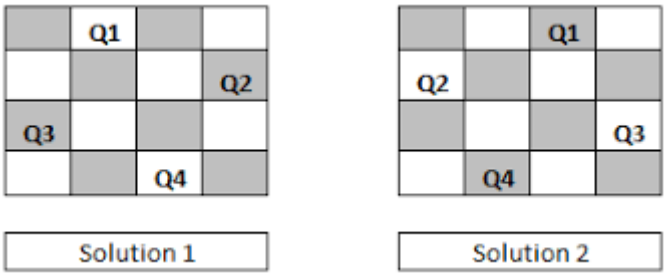
For $n = 7, k = 3$, the only four methods are

$1, 1, 5$; $1, 2, 4$; $1, 3, 3$; $2, 2, 3$.

Eight Queens

Find the number of possible placement of m queens on a $m \times m$ chess board.

For $m = 4$, there are only 2 possible placement:



Hint: Find the relation between this problem and problem 1.10.

Image Deblurring

In image processing, we sometimes need to deal with blurred image to make them clear. In this exercise, you will deblur one image by searching documentations. You need to make make a figure like following figure (the center one is intermediate figure and the right one is the final one)

Hint: for start-up, try `doc *` (for example search deblurring in the search box)