MATH 3341 — Fall 2021

Lab 03: Functions and Control Flows

If you haven't downloaded and unzipped `Math.3341.zip`. Download and unzip it under `H:` (H Drive if you are working on the Remote Lab). Change the current working directory by typing `cd H:\Math.3341\Math.3341.Lab.03` in the Command Window, and type `edit lab_03_script` in the Command Window to edit `lab_03_script.m`.

1 Anonymous Functions

(a) Define an anonymous function `rowSums` which calculates the row sums of a matrix of any dimension. Then define `magicMat5` and `magicMat7` to be a 5×5 and a 7×7 magic square matrix, respectively. Compute `magicMat5RowSums` by calling `rowSums(magicMat5)`, and compute `magicMat7RowSums` by calling `rowSums(magicMat7)`.

(b) Define anonymous functions `f` and `g`, where $f(x) = x\ln(x)$ and $g(y) = ye^y$. Create another anonymous function `h` by composing `f` and `g`, i.e., $h(z) = g(f(z))$. Use `linspace` to define a *column vector* `z`, of which the range is from 1 to 5 with 11 entries. Evaluate function `h` at `z`, and assign the result to `hz`.

(c) Define an anonymous function `matProd` for calculating the product of two matrices, that is, $matProd(A, B) = ABB^TA^T$. Define `A` and `B` using `colon`, `reshape` and `transpose` as follows,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 7 & 10 & 13 & 16 \\ 8 & 11 & 14 & 17 \\ 9 & 12 & 15 & 18 \end{bmatrix}.$$

Store the result of `matProd(A, B)` to `matProdAB`.

(d) Define an anonymous function `p`, where

$$p(x) = \begin{cases} x^3 & \text{if } x < -1, \\ x & \text{if } -1 \le x \le 1, \\ x^2 & \text{if } x > 1. \end{cases}$$

Next visualize $p(x)$ using `fplot` on [-2, 2] (Use `help fplot` for more details about `fplot`). Then run `print(gcf, '-dpng', 'lab_03_1d.png')` to save the plot to a `.png` file.

2 Function Files

Algorithm 1: Recursive Factorial

Function factorialRecursive(n)**:**
  Input: n: an nonnegative integer
  Output: f: $n!$
1 **if** $n = 0$ **then**
2 $f \leftarrow 1$;
3 **else**
4 $f \leftarrow n \times$ factorialRecursive$(n-1)$;
5 **end**
end

Algorithm 2: Iterative Factorial

Function factorialIterative(n)**:**
  Input: n: an nonnegative integer
  Output: f: $n!$
1 $f \leftarrow 1$;
2 **for** $i \leftarrow 1$ **to** n **do**
3 $f \leftarrow f \times i$;
4 **end**
end

(a) Create a function file `factorialRecursive.m` to implement the pseudocode in Algorithm 1.

(b) Create a function file `factorialIterative.m` to implement the pseudocode in Algorithm 2.

(c) In the script file `lab_03_script.m`, use a for-loop to calculate $n!$ where $n = 1, \ldots, 20$ by calling the above two function files as follows

```
fprintf('%2s %20s %20s\n', 'n', 'factorialRecursive', 'factorialIterative');
for n = 1:20
    f1 = factorialRecursive(n);
    f2 = factorialIterative(n);
    fprintf('%2d %20d %20d\n', n, f1, f2);
end
```

3 Application: Real-Life Problems

(a) Create a function file `dayOfWeek.m` to calculate day of week of a specific date. It is known that January 1st, 1970 is Thursday. In the script file `lab_03_script`, calculate the day of week for 01-07-1970, 03-07-1970, 03-08-1971, 08-08-1988, 09-09-1999, 02-10-2021. Here is the suggested syntax for the function: `d = dayOfWeek(year, month, day)`. For example, calling `dayOfWeek(1970, 1, 1)` should return `'Thursday'`. You may use the provided function file `isLeapYear.m`, use `help isLeapYear` for more information. You should use both `if` and `swtich` statements.

Hint: Given that 01-01-1970 is a Thursday, your job is to determine the day of week of a specific date. You can compute the total number of days elapsed since 01-01-1970, disregard the number of weeks past between the dates and add the remainder to the day of week of 01-01-1970 to obtain the day of week of the given date. For example,

- 01-02-1970: 1 day elapsed since 01-01-1970, Thursday + 1 day = Friday. Therefore, 01-02-1970 is a Friday;
- 01-08-1970: 7 days elapsed since 01-01-1970, Thursday + 7 days = Thursday + 1 week = Thursday. Therefore, 01-08-1970 is a Thursday;
- 03-02-1970: 31 (number of days in Jan) + 28 (number of days in Feb) + 2 (number of days past in March) $-1 = 60$ days elapsed since 01-01-1970, Thursday + 60 days = Thursday

+ 8 week + 4 days = Thursday + 4 days = Sunday + 1 day = Monday. Therefore, 03-02-1970 is a Monday;

(b) Shop A is selling a beverage which costs $2 per bottle and there are rules for promotional sales for the beverage:

- You can exchange 4 caps for 1 full bottle of the same beverage for free;
- You can exchange 2 empty bottles (without caps) for 1 full bottle of the same beverage for free.

You have $10 in your pocket. What is the maximum number of bottles of the beverage you can get? Solve this question by writing a function `maxBeverageBottles`.

- The function can be called as below:
  `maxBeverageBottles(money, pricePerNewBottle, capsPerNewBottle, emptyBottlesPerNewBottle)`
  which returns the maximum number of bottles you can get with `money`.
- `money` is the amount of money available;
- `pricePerNewBottle` is the unit price for buying a new bottle;
- `capsPerNewBottle` is the number of caps needed for exchanging 1 free bottle;
- `emptyBottlesPerNewBottle` is the number of empty bottles needed for exchanging 1 free bottle.

Then in the script file, store the result of calling `maxBeverageBottles(10, 2, 4, 2)` to `maxBottles1`. What if you have $1000 in your pocket, the beverage still costs $2/bottle, but 5 caps/bottle or 3 empty bottles/bottle for exchanging a new free bottle, how many bottles can you get? Store the result to `maxBottle2`. You may find `floor` and `mod` useful.

Hint: For each new bottle you get, you have a new pair of cap and empty bottle (after consumed). So you can get new bottles until no enough money/caps/empty bottles.

Before proceeding, make sure you suppress the output in the function files and do NOT suppress the output in the script file. In the Command Window, enter the command `diary('lab_03_output.txt')`, run the script file `lab_03_script.m`, then type `diary off` to store the output to `lab_03_output.txt`. Then upload the script file `lab_03_script.m`, plot file `lab_03_1d.png`, output file `lab_03_output.txt`, and function files `factorialIterative.m`, `factorialRecursive.m`, `dayOfWeek.m`, `maxBeverageBottles.m` to the folder `src` on Overleaf.