

MATH 3341 — Spring 2020

Lab 03: Functions and Control Flows

Download `Math.3341.Lab.03.zip`, unzip it by following the Windows Instructions on WyoCourses. Change the current working directory of MATLAB to the unzipped folder, and type `edit lab_03_script` in the Command Window.

1 ANONYMOUS FUNCTION

- (a) Define an anonymous function `f`, where $f(x) = \sin x - \frac{x^3}{9}$.
- (b) Define a composite function `g`, where $g(y) = e^{\cos(f(y))+y^4}$.
- (c) Use `linspace` to define a *column vector* `t`, of which the range is from -1 to 1 with 10 entries.
- (d) Evaluate function `g` at `t`, and assign the result to `h`.

2 CREATING FUNCTION FILES

Algorithm 1: Recursive Factorial	Algorithm 2: Iterative Factorial
Function factorial_recursive(<i>n</i>): Input: Integer <i>n</i> Output: <i>n</i> ! 1 if <i>n</i> = 0 then 2 <i>f</i> ← 1; 3 else 4 <i>f</i> ← <i>n</i> × factorial_recursive(<i>n</i> − 1); 5 end 6 return <i>f</i> ; end	Function factorial_iterative(<i>n</i>): Input: Integer <i>n</i> Output: <i>n</i> ! 1 <i>f</i> ← 1; 2 for <i>i</i> ← 1 to <i>n</i> do 3 <i>f</i> ← <i>f</i> × <i>i</i> ; 4 end 5 return <i>f</i> ; end

- (a) Create a function file `factorial_recursive.m` to implement the pseudocode in Algorithm 1.
- (b) Create a function file `factorial_iterative.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 $n = 1, \dots, 20$ by calling the above two function files as follows

```
1 fprintf('%2s %20s %20s\n', 'n', 'factorial_recursive', 'factorial_iterative');
2 for n = 1:20
3     f1 = factorial_recursive(n);
4     f2 = factorial_iterative(n);
5     fprintf('%2d %20d %20d\n', n, f1, f2);
6 end
```

3 SOLVING PRACTICAL 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 01-01-1970, 07-17-1975, 12-31-1999, 02-12-2020. 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.
- (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 lids for 1 full bottle of the same beverage for free;
 - You can exchange 2 empty bottles for 1 full bottle of the same beverage for free.

Now you have \$10 available, write a function `totalBottles` to calculate the maximum number of bottles of the beverage you can buy. The suggested syntax for calling the function is `total_bottles = totalBottles(money, price_per_bottle, lids_per_bottle, bottles_per_bottle)`, where `money` is the amount of money available, `price_per_bottle` is the unit price, `lids_per_bottle` is the number of lids needed for exchanging 1 free bottle, `bottles_per_bottle` is the number of empty bottles needed for exchanging 1 free bottle. Then in the script file, store the result of calling `totalBottles(10, 2, 4, 2)` to `total_bottles`. You may find `floor` and `mod` useful.

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`, output file `lab_03_output.txt`, and function files `factorial_iterative.m`, `factorial_recursive.m`, `dayOfWeek.m`, `totalBottles.m` to the folder `src` on Overleaf.