

CMPT125 D101-102, Fall 2025  
Lab exam - Thursday, 9:30am - 10:20am  
November 20, 2025  
You need to implement the functions in ***labexam.c***.  
Submit only the **.c** file to Coursys  
Coursys Assignment - **Lab Exam D101-D102**.

You have 50 minutes to solve all 3 problems.  
The maximal score is 20 points.

The exam will be graded both **automatically** and by **reading your code**.  
You can run your code using  

```
>> make  
>> ./run_test
```

**Correctness:** Make sure that your code compiles without warnings/errors, and works as expected.

**Readability:** Your code should be readable. Add comments wherever necessary. If needed, write helper functions to break the code into small, readable chunks.

**Compilation:** Your code **MUST** compile in CSIL with the Makefile provided. If the code does not compile in CSIL, the grade on the assignment is 0 (zero). Even if you can't solve a problem, make sure it compiles.

**Helper functions:** If necessary, you may add helper functions to the .c file.

**main() function:** do not add main(). Adding main() will cause compilation errors, as the main() function is already in the test file.

**Using printf()/scanf():** Your function should not have any unnecessary printf() statements. They may interfere with the automatic graders.

**Warnings:** Warnings during compilation will reduce points. More importantly, they indicate that something is probably wrong with the code.

**Testing:** An example of a test file is included. Your code will be tested using the provided tests as well as additional tests. You are *strongly encouraged to write more tests* to check your solution is correct, but you don't need to submit them.

### Question 1 [6 points]

Write a function that gets an array of  $n$  strings and returns the total number of digits (0...9) in all strings. For example,

- If  $ar = ["Hello", "", "AB2C", "Hi!7", "123", "-- \&"]$ , the function returns 5.

```
// the function gets an array of length n of strings
// returns the total number of digits (0...9) in all strings
int count_digits(char** ar, int n);
```

### Question 2 [7 points]

Write a function that gets an array of ints of length  $n > 0$ , and a function  $f$ .

It keeps  $ar[0]$  as is

It computes  $f(ar[1])$  and stores the answer in  $ar[1]$

It computes  $f(f(ar[2]))$  and stores the answer in  $ar[2]$ , and so on...

It applies  $f(f(f(f(...))))$   $k$  times on  $ar[k]$ , and stores the answer in  $ar[k]$  for all  $k=0...n-1$ .

For example,

- If  $ar = [0, 0, 0, 0, 0, 0]$ , and the  $f = \text{plus1}$ , then after  $\text{map\_k\_times}$  returns we have  $ar = [0, 1, 2, 3, 4, 5]$ .

```
// the function gets an array of ints of length n, and a function f
// it applies f(f(f(f(...)))) k times on ar[k], and stores the answer in ar[k]
// for all k=0...n-1
void map_k_times(int* ar, int n, int(*f)(int));
```

### Question 3 [7 points]

Write a function that gets a linked list of ints, and returns the average difference between pairs of consecutive nodes

For example,:

- On input 3 -> 2 -> 8 -> 10 -> 5

The differences are 1, 6, 2, 5. Their average is  $(1+6+2+5)/4 = 14/4 = 3.5$

Note that all differences are taken in absolute value

See the file `lib/LL.h` for the functions you can use.

```
// the function gets a Linked List of ints
// and returns average difference between pairs of consecutive nodes
// if size of list <= 1, the function returns -1.
float average_diff(const LL_t* list);
```