

## CMPT125, Fall 2021

### Lab exam - Version 4

Wednesday, November 17, 2021, 4:30pm-5:20am

You need to implement the functions in ***labexam.c***.

Submit only the **.c** file to Canvas

Canvas Assignment - Lab Exam 16:30-17:20.

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_v4
```

**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 have no 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 [7 points]

Write a function that gets two strings, *str1* and *str2*, and searches for *str2* inside *str1*, and returns the index of the first appearance of *str2* in *str1*. If *str* is not in *str*, the function returns -1. For example:

- `str_find("Hello world", "llo")` should return 2.
- `str_find("ABCDEFD", "DE")` should return 3.
- `str_find("ABCDEFD", "BCF")` should return -1.

```
// finds the first appearance of str in str1
// returns the index of the beginning of the first appearance
// if str2 is not in str1, returns -1
int str_find(const char* str1, const char* str2);
```

### Question 2 [6 points]

Write a function that gets a 2-d array of ints *arr*[height][width], and returns an array *SUM* of length height such that *SUM*[i] contains the sum of all values in the *i*'th row of the array. For example

- On input  $\{\{ 1, 2, 3\},$   
           $\{4, 7, -2\},$   
           $\{9, 10, 11\},$   
           $\{1, 5, 4\}\}$ ;  
the function returns  $\{ 6, 9, 30, 10\}$

```
// returns array SUM with SUM[i] = sum in the i'th row
int* sum_row(int height, int width, const int ar[height][width]);
```

### Question 3 [7 points]

Write a function that gets a linked list of ints, and reverses it. For example

- on input 1 -> 2 -> 3 -> 4, after the function finishes the execution, the list becomes 4 -> 3 -> 2 -> 1
- If the list has one element, then it doesn't change
- If the list is empty, then it doesn't change

You may use the data fields in the struct and the functions provided in *LL.h* and *LL.c*.

```
// reverses a linked list
void LL_reverse(LL_t* list);
```