

CMPT125 D105-106, Fall 2025
Lab exam - Tuesday, 10:30am - 11:20am
November 18, 2025
You need to implement the functions in ***labexam.c***.
Submit only the **.c** file to Coursys
Coursys Assignment - **Lab Exam D105-D106**.

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 a string and two parameters $0 \leq \text{start} \leq \text{end} < \text{strlen}(\text{str})$. The function reverses the substring `str[start...end]`

For example,

- On input `str = a = "012345678ABC"`, `start=4`, `end=7`, the string will become `"012376548ABC"`.

```
// the function gets a string and two parameters start<=end
// and reverses the substring str[start...end].
// Assumption: 0 <= start <= end < strlen(str)
void reverse_str(char* str, int start, int end);
```

Question 2 [7 points]

Write a function gets an array of ints of length `n`, and a predicate. It removes all elements in the array satisfying `pred(arr[i])==false`, and keeps only the elements in the array satisfying `pred(arr[i])==true`. If an element is removed, the remaining elements are shifted to the left. The function returns the number of elements in the remaining array. For example,

- On input `a = [1,2,3,6,8,7,9]`, and `pred = is_even`, the function returns 3, and the array `a` becomes `[2,6,8,*,*,*]`, where `*` can be anything

```
// the function gets an int a, a function f, and an int n.
// it keeps only the elements in the array satisfying pred(arr[i])==true,
// pushing them to the left if needed.
int filter_array(int* a, int n, bool(*pred)(int));
```

Question 3 [7 points]

Write a function that gets a linked list of ints and a predicate, and returns the sum of all items in the list satisfying `pred(item)==true`.

For example,:

- On input `3 -> 2 -> 7 -> 8 -> 10 -> 5` and `pred = is_even` the function returns `2+8+10 = 20`.

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

```
// the function gets a Linked List of ints and a predicate
// and returns the sum of all items in the List satisfying pred(item)==true
int LL_sum_cond(LL_t* Lst, bool(*pred)(int));
```