

CMPT125, Spring 2022
Lab exam - D203-D204

Tuesday, March 15, 2022, 10:30am-11:20am
You need to implement the functions in **labexam.c**.
Submit only the **.c** file to Coursys
Coursys Assignment - Lab Exam 10:30-11:20.

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
```

Submit only labexam.c: Please make sure to submit the file to the *correct section* in Coursys.

Correctness: Your file must compile without warnings/errors, and work 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 *labexam.c* file.

main() function: do not add main() to labexam.c. 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 [6 points]

Write a function that gets a string and returns the number of substrings consisting of hyphens only. For example:

- `count_hyphens_substrings("--abc-XYZ")` returns 2.
- `count_hyphens_substrings("ab***c-----X-Y-Z-")` returns 4.
- `count_hyphens_substrings("abc*XYZ")` return 0.

```
// the function gets a string
// and returns the number of substrings consisting of hyphens only.
int count_hyphens_substrings(const char* str);
```

Question 2 [7 points]

Write a function that gets a linked list and a boolean function `pred`. The function returns the number of nodes in the linked list for which `pred(node->data)==true`. For example:

- `countif([1, 8, 3, 3, 8, 1, -4, -6] , is_odd)` returns 4.
- `countif([-1, 20,-2, 1, -8] , is_positive)` returns 2.

See `lib/LL.c` and `lib/LL.h` for details.

```
// gets a Linked List of ints and a predicate
// returns the number of nodes in the list for which pred(node->data)==true
int countif(LL_t* list, bool(*pred)(int));
```

Question 3 [7 points]

Write a function that gets a stack `orig`, and returns a new copy of the stack. See `lib/stack.c` and `lib/stack.h` for details.

```
// gets a stack orig, and creates a new stack with the same content
// when the function returns, orig needs to be in its initial state
stack_t* stack_copy(stack_t* orig);
```