

CMPT125, Spring 2022  
Lab exam - D207-D208

Tuesday, March 15, 2022, 12:30pm-1:20pm  
You need to implement the functions in **labexam.c**.  
Submit only the **.c** file to Coursys  
Coursys Assignment - Lab Exam 12:30-1:20.

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 rotates it to the right by one char. For example:

- If we have `char* str="ABCDEFGH"`, then after applying `str_right_rotate(str)`, `str` becomes `"HABCDEFG"`.

```
// the function gets a string
// and rotates the characters of the string to the right by 1
void str_right_rotate(char* str);
```

### Question 2 [7 points]

Write a function that gets `ar` - an array of `int` of length `n`, and a function `foo()`.

The function returns `ret` - a new array of length `n`, where `ret[i] = foo(ar[i])`. For example:

- `map_to_new_array( [8,1,2,-6], n=4, minus_one)` returns `[7,0,1,-5]`.
- `map_to_new_array( [1,2,-1], n=4, square)` returns `[1,4,1]`.

```
// the function gets an array of ints of length n, and a function foo
// and returns a new array (on the heap) of the same length
// where ret[i] = foo(ar[i]) for all i
int* map_to_new_array(const int* ar, int n, int(*foo)(int));
```

### Question 3 [7 points]

Write a function that gets two pointers to stacks of `ints` and swaps their content.

If both pointers point to the same stack, the function doesn't do anything.

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

```
// gets two stacks, and swaps their content
void stacks_swap(stack_t* s1, stack_t* s2);
```