

CMPT125, Spring 2022
Lab exam - D101-D102

Thursday, March 17, 2022, 09:30am-10:20am
You need to implement the functions in **labexam.c**.

Submit only the .c file to Coursys
Coursys Assignment - Lab Exam 09:30-10:20

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 an array of ints of length n , such that each element is between 0 and 9, and sorts the array. For example:

- On input [1,4,1,2,2,9] after the function returns the array should be [1,1,2,2,4,9].

```
// the function gets an array of ints of length n and sorts the array
// all numbers in ar are guaranteed to be between 0 and 9 (0 and 9 included)
void sort_digits(int* ar, int n);
```

Question 2 [7 points]

Write a function that gets **a,b** two arrays of ints of length n , and returns a new array such that $ret[i]=\max(a[i], b[i])$. For example:

- `max_arrays([8,1,2,-6], [0,1,4,-2], n=4)` returns [8,1,4,-2].

```
// the function gets two arrays, a and b, of length n
// and returns a new array (on the heap) of the length n
// where ret[i] = max(a[i], b[i])
int* max_arrays(int* a, int* b, int n);
```

Question 3 [7 points]

Write a function that gets a queue of ints, and a boolean function `pred`, and returns the number of elements in the queue such that `pred(element)==true`.
See `lib/queue.c` and `lib/queue.h` for details.

```
// gets a queue of ints and a predicate
// counts the number of elements in the queue satisfying pred(element)==true
// when the function returns the queue needs to be in its initial state
int queue_countif(queue_t* q, bool (*pred)(int));
```