CMPT125, Spring 2023
Lab exam

Monday, March 20, 2023, 2:30pm-3:20pm

You need to implement the functions in ***labexam.c***.
Submit only the ***.c*** file to Coursys
Coursys Assignment - **Lab Exam D201-D202 Mon 2:30pm**

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.

Good luck!

Question 1 [6 points]

Write a function that gets an array of ints of length n, and returns the minimum and the maximum of the array. The struct min_max is defined in labexam.h.

```
// gets an array of ints
// and returns the minimum and the maximum of the array
// the returned values are stored using the struct min_max
min_max get_min_max(int* arr, int n);
```

Question 2 [7 points]

Write a function that gets a 2d array, and returns an checks if the array contains two columns that are equal

```
// gets a 2d array of ints
// returns true if the array contains two equal columns
bool has_equal_columns(int height, int width, const int ar[height][width]);
```

Question 3 [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));
```