

CMPT125, Spring 2023
Lab exam

Monday, March 20, 2023, 3:30pm-4:20pm

You need to implement the functions in ***labexam.c***.
Submit only the **.c** file to Coursys
Coursys Assignment - **Lab Exam D203-D204 Mon 3:30pm**

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 checks if the numbers are sorted in non-decreasing order. For example

- on input [1,2,2,5,8] it needs to return true
- on input [2,3,1,8] it needs to return false
- on input [2,2] it needs to return true
- on input [2,-2] it needs to return false

```
// gets an array of ints
// checks if the numbers are sorted in non-decreasing order
bool check_non_decreasing(int* arr, int n);
```

Question 2 [7 points]

Write a function that gets an array of ints of length n , and returns a new array that contains the same numbers in the reverse order.

```
// gets an array A of length n
// returns a new array that contains all entries of A in the reverse order
int* rev_arr(const int* A, int n);
```

Question 3 [7 points]

Write a function that gets an array of strings of length $n > 0$, and returns the number of distinct strings in the array. For example:

- On input = ["Hello", "Hello", "hi", "Hi", "Hello", "Hi"] the function returns 3.
- On input = ["A", "B", "C", "D", "E"] the function returns 5.
- On input = ["c++", "c++", "c++", "c++", "c++"] the function returns 1.

Two strings are distinct if `strcmp(str1, str2) != 0`.

```
// the function gets an array of strings of length  $n > 0$ 
// and returns the number of distinct strings in the array
int count_distinct(const char** ar, int n);
```