

CMPT125, Spring 2022  
Lab exam - D205-D206

Tuesday, March 15, 2022, 11:30am-12:20pm  
You need to implement the functions in **labexam.c**.  
Submit only the **.c** file to Coursys  
Coursys Assignment - Lab Exam 11:30-12:20.

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 computes the sum of all digits in the string. For example:

- `sum_of_digits("--ab1c-X3Y6Z")` returns `1+3+6=10`.
- `sum_of_digits("--a0b1X!@#201")` returns `0+1+2+0+1=4`.
- `sum_of_digits("1234")` returns `1+2+3+4=10`.
- `sum_of_digits("abcd")` returns `0`.

```
// the function gets a string
// and computes the sum of all digits in the string
// all non-digit chars are ignored
// you may assume the sum fits into an int
int sum_of_digits(const char* str);
```

### Question 2 [7 points]

Write a function that gets an array of int of length `n`, and returns a new array of length `2n`, where each value of the input is duplicated. For example:

- `duplicate_values([8,1,2,-6], n=4)` returns `[8,8,1,1,2,2,-6,-6]`.
- `duplicate_values([1,2,-7,1,6], n=5)` returns `[1,1,2,2,-7,-7,1,1,6,6]`.

```
// the function gets an array of ints of length n
// and returns a new array (on the heap)
// where each value of the input is duplicated
int* duplicate_values(const int* ar, int n);
```

### Question 3 [7 points]

Write a function that gets a linked list, and a boolean function `pred`. The function removes all nodes for which `pred(node->data)==false`, and returns the number of removed elements. See `lib/LL.c` and `lib/LL.h` for details.

```
// gets a Linked List of ints and a predicate
// and keeps in the list only the nodes satisfying pred(node->data)==true
// all nodes not satisfying pred are removed from the list
// returns the number of removed elements
int LL_filter(LL_t* list, bool (*pred)(int));
```