

CMPT125, Spring 2022  
Lab exam - D107-D108

Thursday, March 17, 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 the function that gets an array of strings of length  $n$ , and returns the concatenation of all strings (on the heap). For example:

- `concat_all(["AB", "CD", "EFG"], n=3)` returns "ABCDEFGF".

```
// the function gets an array of strings of length n  
// and returns one string containing the concatenation of all strings  
char* concat_all(const char* arr[], int n);
```

### Question 2 [7 points]

Write a function that gets a non-negative number, and checks if it is a palindrome. For example, 2, 9009, 12321 and 761167 are palindromes, but 123, 766, 1232 are not.

```
// gets a non-negative number, and checks if it is a palindrome  
// for example: 2, 9009, 12321 and 761167 are palindromes  
bool is_palindrome(unsigned int n);
```

### Question 3 [7 points]

Write a function that gets a linked list and a non-negative int  $k$ . It removes the last  $k$  elements from the list, and returns the sum of the values in the removed nodes

Assumption: the length of the list  $\geq k$ .

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

```
// gets a linked list and a non-negative int k  
// removes the last k elements from the list  
// returns the sum of the values in the removed nodes  
// assumption: the length of the list  $\geq k$   
int LL_remove_last_k(LL_t* list, unsigned int k);
```