0512-1820 Fall 2024
Home Assignment #(2)

Yarden Haskin: yardenhaskin@tau.ac.il

Due date: (29/11/2024)

In this assignment you will get hands-on experience with (recursion and functions) in the C programming language.

Submission Guidelines

- Due date is (06/12/2024)
- Submission file is: **hw(2)_ID1_ID2.zip**

  The zip should include the following files:

  - **ex1_ID1_ID2.c**
  - **ex2_ID1_ID2.c**

  E.g. for a pair of students with IDs 123456789 and 987654321 the zip file should be named:

  hw(2)_123456789_987654321.zip

  And for example, the <u>first</u> source file should be named

  ex1_123456789_987654321.c

  This is a mere example. Do not forget to zip all of the required files as mentioned above.

- Please use the standard C libraries only! no other libraries should be installed or used unless you have been specifically instructed.
  `<math.h>` or any other header file except `<stdio.h>` are not allowed in this home assignment.
- For any issues & questions you can use the forum in moodle, consult with your peers and also use google.

- **Warning 1**: If your code doesn't compile you will get 0, regardless of the amount of work you've put into coding.
- **Warning 2**: Do not cheat, or use any automatic code generator to complete this home work! It's for your own good. Caught cheaters will be **punished**!

Suggested workflow

1. Generate new **git** repository for this home assignment / Add a new directory to your existing HW git repository.
2. If there are any attached files, **download them** from moodle into your repository.
3. **Open & read the given files** for this home assignment.
4. **Read this entire document** before writing a single line of code.
5. Write some basic **tests** to make sure your code will work (TDD).
6. Let the **coding** begin!
   Don't forget to **commit & push your progress** in git for version control & collaboration.
7. Make sure your **code compiles** in the testing environment.
8. **Add more tests** with all of the corner cases you could think.
9. Make sure your **code runs properly** and correctly, and that all of your tests pass.
   Debug your code and fix it accordingly (you might find "rubberducking" pretty useful).
10. **Re-read this document** to make sure you haven't forgotten anything.
11. **Check the moodle for any updates** regarding this assignment in the Q&A forum and in the Announcements forum.
12. **Zip your code** according to the submission guidelines above.
13. **Unzip your code** and repeat steps 7 & 9 to make sure everything is OK.
14. **Submit the zip file** to moodle.
15. Congratulations you have **completed** the home assignment!

Good luck!

Exercise 1

Problem Explanation
A palindrome is a string that reads the same forwards and backwards, ignoring case and non-alphanumeric characters.

Examples:

"radar" → Palindrome
"A man, a plan, a canal, Panama" → Palindrome
"Hello" → Not a Palindrome
This exercise requires you to:

Write a recursive function to check if a string is a palindrome.
Write an iterative function for the same task.

Exercise 2

Generate Subsequences
Problem Explanation
A subsequence is a subset of characters from a string that appear in the same order as in the original string, but not necessarily consecutively.

Examples:

Original String: "abc"
Subsequences: ["", "a", "b", "c", "ab", "ac", "bc", "abc"]

* Note that skeleton scripts are provided for each exercise. You must use these scripts and implement your code only in the designated sections: // TODO: Implement the logic for recursive palindrome check