Homework Submission Instructions

Read carefully!!

Homework and the final grade

- **There will be 5 homework assignments.**
- Every program should be written in C, in a Unix-like environment, unless stated otherwise.
- Homework will be graded on VirtualBox running Lubuntu.
- **The moodle forum** is part of the course and the homework requirements. You are expected to follow it and **all the** answers provided within it.
- You will have two weeks to solve each assignment.
- **Late submissions**
    - Extension requests must be made **before** the **original** due date.
        - Extensions will only be granted for miluim or medical/personal emergencies
    - Extension requests must be accompanied by a written verification.
    - Extensions are **not granted** for *any other reason*. You can avoid any problem by having the assignment ready by its **original** due date, and carefully checking your work **BEFORE** you submit.
- You fail the assignment **if you did not submit on time** without a note **in advance.**

Appeals

- Appeals should be sent directly to the grader **within 5 days** after publishing grades.
- You may **only** appeal regarding errors in grading. You may not appeal regarding amount of points deducted, or anything similar – the grader will ignore such appeals.
- Appeals will be answered within a week. **Please be patient**.
- You can appeal the appeal by forwarding all details to your TA, who will forward it to the lecturer **if needed**.

Online submission

- The exercises will be submitted through moodle.
- The submission should be a single zip file.
- Submit zip files ***only***! No rar, rar with .zip extension, 7z, etc.
- The name of the submitted zip should contain your **9-digit** ID number along with the exercise number. For ID 012345678 and exercise #3, name it ex3_012345678.zip **EXACTLY!**
- Use the *zip* command to create a zip file, e.g., `zip 012345678_ex1.zip cipher.c script.sh`
- Submit only the zip file, with all files flat inside it - **no subfolders inside!** All files in the root!
- File and directory names are **case-sensitive**.
- Submissions must compile with a simple GCC command. No flags unless stated otherwise (e.g., `gcc -o cipher cipher.c`)
- Text explanations, where required, are in **English only in an answers.txt file**.
- No spaces or Hebrew characters in file names.
- Do NOT submit binaries, temporary files or folders, mac addendums, etc.
- **Don't code in Windows/Mac.** Conversion errors are on you and will result in deductions!
- Check your submission **carefully** before submitting. Small errors (such as forgetting to comment-out a line, submitting old code, compile error, etc.) can result in large point deductions and failing grades, and any appeals will be denied.
- **Again** – **double and triple-check** your submission. Make sure everything is correct, file names are exact, there are no folders **at all** in the zip file, compilation succeeds, file is a zip, etc.
- **Points will be deducted for errors in submission!** No appeals are allowed.

General guidelines

- A working submission is not enough! Correctness is only the basic requirement, and you are graded for much more than that, such as proper memory and HD utilization, safe system calls and proper error handling, **and more**.
- Check your submission on a **fresh** VM to make sure it *really* works!
- Read no input from *stdin* ($getch()$, etc.) unless specifically required.
- Use **only** system calls where possible, and **only** those that were covered in the recitations.
- Check the return values of **all** system calls (except `close` and `closedir`).
- Don't use `assert()` or other unsafe calls. Avoid using the *C* standard library (`printf` is ok).
- Your code should compile without warnings (but no need for *–Wall* flag).
- Make sure your program terminates and doesn't hang. This is **VERY** important!
- **All files** created by your code, in any case, should be given proper permissions (*can be 0777*).
- No need to check correctness of arguments or input, **unless stated otherwise**.
- Don't use more than **1MB** contiguous memory. If you need more then **you are doing something wrong**.
- Limit yourself to **1GB total RAM**. You should avoid getting *even close* to this limit.
- No need to exit cleanly on errors; however, you **must** output a proper error message, describing the logical part **and** the system call that failed, as well as the specific error message provided by the OS (i.e., $strerror(errno)$). Otherwise, it is **not** a proper error message.
- **Points will be deducted for failing to meet these guidelines!**