Programming for Data Science (Full exam 03/09/2024)

Upload the solutions to the programming exercises to the following link:
https://evo.di.unipi.it/student/courses/16/exams/x34dKq3

Exercise 1. (Math, on paper)

A. Complete the following definitions for (sub)sets of the Natural numbers, which include, respectively, only odd numbers and even numbers:

Set Odd includes only odd numbers. In the definition use the congruence relation modulo
Odd = {x ∈ **N** | ………………….. }

Set Even includes only even numbers. In the definition use the a divides b (a | b) relation
Even = {x ∈ **N** | ………………………..}

B. Let $M = \begin{bmatrix} 1 & 1 \\ a & b \end{bmatrix}$, show how you find a and b such that $M^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

C. Let R ⊂ {1,2,3,4,5} × {1,2,3,4,5} be a relation defined as follows: R = {(1,3), (2,2), (2,5), (3,1), (3,2), (3,5), (4,4), (5,2), (5,4), (5,5)}

Is R symmetric? If the answer is positive, motivate it. If not, show a counterexample.

Exercise 2. (Python) Create a Python program that takes in input a list **l** of **n** integers from the user, with **n** to be provided by the user at the beginning of the program. Then, implement and invoke the following functions:

1. **sort(l)**: sort the list in ascending order. You should write the sort implementation on your own! Bonus: sort the list in place, i.e., without exploiting additional space!
2. **search(l, x)**: search for the number **x** within the list **l**, returning its index if **x** is found, False otherwise. Bonus: exploit the feature of working on a sorted list! E.g., search([1,4,5], 4) -> 1, search([1,4,5], 3) -> False
3. **sortedInsert(l, x)**: insert **x** into the sorted list **l** and return the resulting sorted list. Bonus: do not invoke the sort method after the insert! E.g., addNumber([1,4,5], 2) -> [1,2,4,5]
4. **sortedDelete(l, x)**: if **x** is in the list **l**, delete it and return the sorted list. Otherwise return the unmodified list **l**. Bonus: do not invoke the sort method after the insert! E.g., sortedDelete([1,4,5], 4) -> [1,5], sortedDelete([1,4,5], 3) -> [1,4,5]

Exercise 3. (C) Create a C program that implements some basic string functionalities. Do not use standard string library functions like strlen, strcpy, or strstr. Instead, implement the necessary operations manually. The program should implement the followings:

1. Read from the user two strings.
2. Implement a function **int count_words(char *text)** that takes a string text as input and returns the total number of words in the string. Assume that words are separated by spaces.
3. Implement a function **void reverse_string(char *text)** that takes a string text as input and returns the reversed string. <u>Bonus: do not use any additional support space, i.e., reverse the string in place.</u>
4. Implement a function **int find_substring(char *text, char *substring)** that takes a string text and a substring and returns the starting index of the first occurrence of the substring in text. Return -1 if the substring is not found.
5. Invoke each implemented function and display the result on the screen.