Programming for Data Science (02/02/2024)

Upload the solutions to the programming exercices to following link:
https://evo.di.unipi.it/student/courses/16/exams/NVO5ZWv

Exercise 1. (Math, solve the exercise on paper) Consider the scenario of implementing a phone book. As an example, imagine you have to store:

"John Doe", 5551234
"Jane Smith", 5555678
"Alice Johnson", 5559876
…..

A. Present, with examples, what a hash table, a hash function, a collision are, in this scenario.
B. Describe a collision resolution technique.
C. What are the properties of the hash function “h(k)” ? Is it injective (aka ono-to-one)? Surjective (aka onto)? Invertible?
D. Use first order logic to formalize that a function f: N→ N is injective.

Exercise 2. (Python) A DNA string is a string built on an alphabet of 4 characters: A, T, G, and C. In bioinformatics a *k-mer* is a substring of *k* characters from a string that is longer than *k*. Write a function **KMer(s,k)** with two parameters: a DNA string and the value of k. Return a dictionary of *k-mer* counts. **BONUS**: store in the dictionary the list of positions from where each *k-mer* starts.

Example: **KMer(GTAGAGTAGT, 3)** → {“GTA”:2, “TAG”:2, “AGT”:2, “AGA”:1, “GAG”:1}
BONUS *Example*: **KMer(GTAGAGTAGT, 3)** → {“GTA”:[0,5], “TAG”:[1,6], “AGT”:[4,7], “AGA”:[2], “GAG”:[3]}

Exercise 3. (C) Write a C program that:

- Asks the user to input a number n greater than 50
- Allocate the memory space needed to store an array of n integers
- Fill the array with randomly generated values between -200 and 300
- Prints the content of the list
- Then, sorts the values in the list, in increasing order. No extra data structures can be used (i.e., do not store the values in an array to sort them) and do not use library functions. The sort should be implemented by your own.
- Prints the content of the sorted list