Programming for Data Science (31/10/2023)

0% of the points are assigned to quality of documentation and/or comments to solutions. Solutions must include tests of executions of the developed functions.

Name files as “<your matricola>_<firstname>_<lastname>_ex1.py” for Exercise 1, and “<your matricola>_<firstname>_<lastname>_ex2.c” for the second exercise.

Upload the TWO files in a folder (named with your student number and your last name) at the following URL: Upload here (access GDrive using your university credentials)

Exercise 1. (Math, on paper)

Consider the following sets:

\[ R = \{ p \in \mathbb{Z} \mid -100 \leq p \leq 100 \} \]
\[ A = \{ m \in R \mid m \text{ is a multiple of 5} \} \]
\[ B = \{ n \in \mathbb{Z} \mid n^2 < 100 \} \]
\[ C = \{ 2x + 2 \mid x \in A \} \]

a) Which is the cardinality of the sets: \( A \cap B; B \cap C; A \cap B \cap C \)?

b) List the elements of the set: \( D = \{ (x, y) \in (A \cap B) \times (B \cap C) \mid x \cdot y \leq 0 \} \)

c) Let’s consider the function: \( f : C \rightarrow \mathbb{Z} \) such that \( f(c) = c + 1 \) for every \( c \) in \( C \). Determine if this function is injective, surjective, or bijective.

Exercise 2. (Python)

Implement the Exercise 1 in Python, according with the definition given in the previous exercise:

1. Define the three sets A, B and C
2. Create the new set D made up of all tuples \((x,y)\), with \( x \in (A \cap B) \) and \( y \in (B \cap C) \), such that \( x \cdot y \leq 0 \)
3. Create a function \( product(s, n) \), taking a set \( s \) of tuples \((x,y)\) and a number \( n \) in input, and producing in output a new set resulting from the multiplication of \( x \), \( y \) and \( n \). Test this function on the D set and a number \( n \) to be read from the user (only once, before the invocation of the function).

Exercise 3. (C)

Write a C program that performs basic string manipulation on a user-entered string. The program should provide the implementation for each of the following operations:

1. Calculate the length of the string (without termination character \'\0\')
2. Reverse the string.
3. Convert the string to uppercase.
4. Check if the string is a palindrome (reads the same forwards and backward).

Prompt the user to input a string and then display the result of each operation. The aforementioned operations should be implemented without exploiting the c string functions.