Programming languages (TC-2006)

Extra points challenge 04

In this challenge, you will practice with the Racket language to implement various functions. Please consider that the purpose of this challenge is to allow you to practice and identify strengths and weaknesses. Then, implement these functions as requested and avoid using any built-in functions that already do what you are requested to implement.

1 `prime?` (25%)

Write a predicate in Racket that checks if a number provided as an argument is prime or not. Recall that a prime number is a natural number greater than 1 with no positive divisors other than 1 and itself. You might find the function `remainder` useful for implementing this function.

2 `sumDigits` (25%)

Write a function in Racket that receives a number and returns the sum of its digits. For simplicity, you can assume that the input to your function will always be a number, so no additional checks are required. You might find the functions `quotient` and `modulo` useful for implementing this function.

3 `xor` (25%)

Prepare a function in Racket that takes two lists as input and returns a list where the i-th element is the result of the `xor` operation on the i-th elements of the two lists given as input. For simplicity, you can assume that both lists have the same length and contain boolean values only.

4 `listToNumber` (25%)

Program a function in Racket that takes a list of integers and returns a numeric value of the list by considering each element in the list as one digit in the resulting number. For this function, assume that the numeric value of the empty list is zero. For simplicity, you can assume that the input to your function will always be a list of numbers with no nested lists, so no additional checks are required. You might find the function `expt` useful for implementing this function.

Deliverables

Prepare an RKT file that contains the functions requested and submit it to Canvas. **Please, do not submit other formats but RKT**. To prepare your RKT file, use the code template distributed along with this document. The template contains some test cases for each function to help you verify that your codes work as requested.

I promise to apply my knowledge, strive for its development, and not use unauthorized or illegal means to complete this activity, following the Tecnológico de Monterrey Student Code of Honor.