Question 1 (25%)

As you have observed, the recursive solution to the computation of *n*th Fibonacci number is quite inefficient, as you compute the same Fibonacci numbers multiple times. Such algorithms can be made efficient by using a technique called *memoization*. The basic idea in memoization is to store the result of some computations in a table, so that when the function is called with some arguments already seen, the result is looked up from the table instead of executing the function again.

Your task in this exercise is to make the recursive Fibonacci solution by using memoization. You are required the results computed so far in a *hash table*. Hash tables are dictionaries that keep *key*,*value* pairs. In your case the keys will be integers and the value corresponding to a certain key *n* will be the *nth* Fibonacci number. In order to learn how hash tables are constructed and managed, please refer to the relevant chapter of one of the books listed on the course website.

You will change your recursive Fibonacci solution in such a way that when called with a certain integer, you will first check whether you have computed this Fibonacci number before by looking at your table, and enter into recursion only in case you do not have the number in the table.

Your procedure will be used exactly as the original one, namely it should be a single argument procedure that takes an integer and return the corresponding Fibonacci number.

Question 2 (25%)

Write a program that takes as input a set of numbers and sorts the numbers with respect to their Collatz lengths in descending order. You can use the built-in sort procedure of `LISP`, please consult the reference books listed on the web-site on how to use the built-in sort.

Question 3 (25%)

Define a three argument procedure `REMOVE-NTH`, which removes every nth occurrence of an item from a list. You must use the techniques of Section 8.

Question 4 (25%)

Define a procedure `NTH-LARGE` that returns the *n*th largest number in a list of numbers. You must use the techniques of Section 8.