אוניברסיטת בר אילן - המחלקה למדעי המחשב

שפות תכנות והידור - 89-310 - חורף תש״פ

מרצה: ד״ר אריאל רוט

תרגיל בית 2

1) **Memoization** is an optimization technique used primarily to speed up computer programs by storing the results of function calls and returning the cached result when the same inputs occur again. Memoization is a specific form of caching that involves caching the return value of a function based on its parameters.
   Implement Memoization in Javascript for the following:
   a) F(n) returns Fibonacci number n.
      F(n)=F(n-1)+F(n-2), F(0)=0, F(1)=F(2)=1
   b) F(n)=n! (F(n)=n*F(n-1), F(0)=1)

2) Implement in JavaScript the high-order function **memoize()** that accepts a function f() as its argument and returns a memoized version of the function f().

3) Use your **memoize()** function of question 2 to give another solution to question 1.b (n!).

4) Generalize **memoize()** to accept functions f(...) with several arguments (you can assume that arguments to f all have distinct string representations).

   Hint for questions 2 & 4: The **memoize**() function creates a new object to use as the cache and assigns this object to a local variable, so that it is private to (in the closure of) the returned function. The returned function converts its arguments array to a string, and uses that string as a property name for the cache object. If a value exists in the cache, it returns it directly. Otherwise, it calls the specified function to compute the value for

these arguments, caches that value, and returns it.

5) Write an HTML and a JavaScript files to implement the following adder that takes two number inputs. It outputs their sum after clicking the Add button: