

- 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) $F(n)=n!$ ($F(n)=n \cdot F(n-1)$, $F(0)=1$)

b) $F(n)$ returns Fibonacci number n .

$F(n)=F(n-1)+F(n-2)$, $F(0)=0$, $F(1)=F(2)=1$

- 2) Implement in JavaScript the high-order function **memoize()** that accepts a function (name it f for example) as its argument, and returns a memoized version of the function f (in this section you can assume f has only one argument).
- 3) Based on your **memoize()** solution function of question 2 (an adaptation might be needed to cope with the recursive call), give another solution to question 1 (1.a and 1.b).
- 4) Generalize **memoize()** to accept a function f as its argument like in question 2, but now, f may have several arguments (you can assume that arguments of f all have distinct string representations).
- 5) Write an HTML and JavaScript files to implement the following adder that takes two number inputs. It outputs their sum after clicking the Add button, the web page should be:

Add two numbers

Number 1 Number 2

5 + 6 = 11

הערות:

1. יש להגיש את התרגיל בקבצי js ו html בלבד ולא בכתב יד, כדלהלן:
ex2_1a.js, ex2_1b.js , ex2_2.js, ex2_3a.js, ex2_3b.js, ex2_4.js,
ex2_5.js, ex2_5.html

2. בראש כל קובץ כנ"ל יש לכתוב בהערה את שם המגישים + ת.ז. (רק אחד מכל זוג/שלשה יגיש קבצים עם פתרון, האחרים יגישו רק קובץ ex2.txt שיש בו רק שמות+ת"ז ללא קוד).

3. את התרגילים יש להגיש לאתר הזה בלבד ולא לשלוח אותם אלי או למייל אחר או למערכת אחרת (אין לשלוח לי פתרונות כדי לקבל ממני בדיקה/הערכת ציון ראשונית, זה תפקיד הבודק/ת לאחר הגשה).

4. כמובן שאין להעתיק.

5. לפני שאתם מגישים תבדקו שאכן כל קובץ מכיל קוד שרץ ונותן את הפלט המבוקש. צריך לכתוב בהערות באופן ברור מה מכיל כל קובץ ואיך להריץ. בכל השאלות חובה להוסיף דוגמאות הרצה בהערות בגוף קובצי ההגשה (אתם ממילא מריצים לבדיקה עצמית שהצלחתם).

6. לגבי שאלה 4: אפשר להשתמש ב arguments. אפשר להשתמש ב Array.prototype.join.call כדי לקבל string שישמש key ל cache , צריך כמובן לחשוב איך לייצר key חד חד ערכי, כלומר שסטים שונים של ארגומנטים יקבלו keys שונים.

7. בהצלחה!