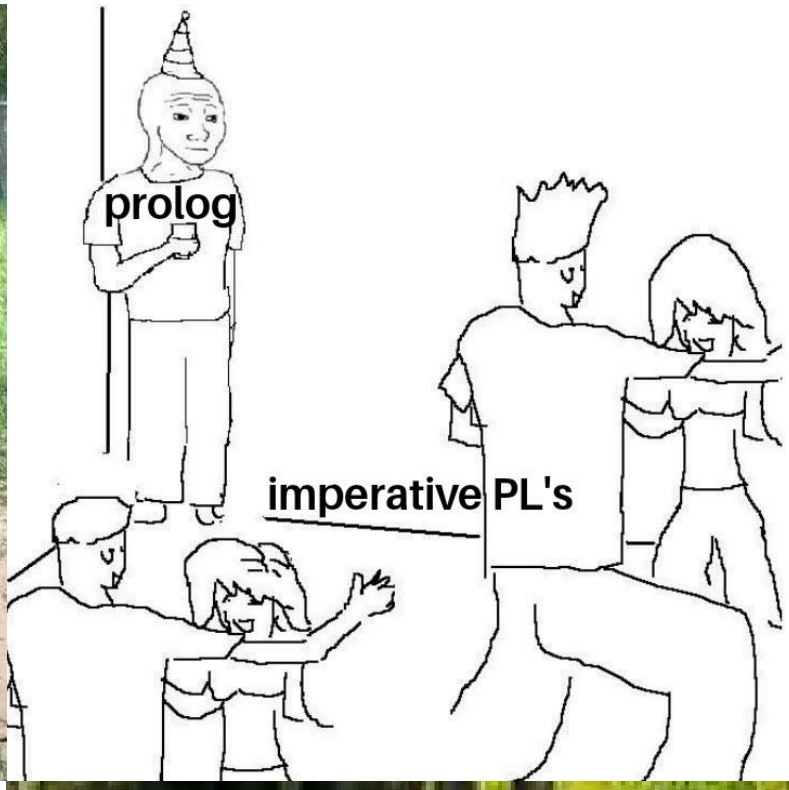
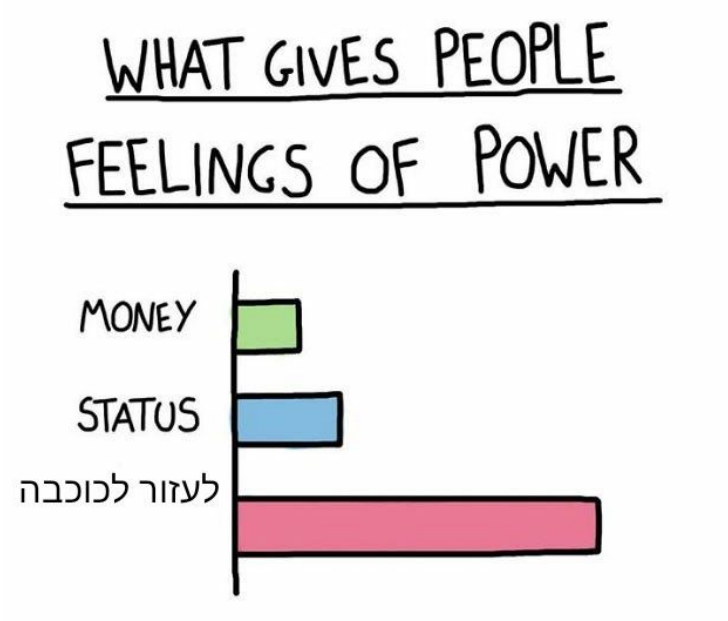
תרגיל בית 6

מגישים: אסף לובטון 209844414

ועדן דמבינסקי 212227888.





@iamnotanartist_

שאלה 1

בתרגיל זה עליכם לכתוב תוכנית **אחת** לפי ה-syntax של שפת C שאותה נריץ באופן **היפותטי** 3 פעמים:

1. בפעם הראשונה, נריץ אותה באופן היפותטי עם אסטרטגיית השערוך EAGER, ועל התוכנית להדפיס את המילה EAGER. כלומר, אם בשפת C אסטרטגיית השערוך הייתה EAGER, אז התוכנית הייתה מדפיסה את המילה EAGER.
2. בפעם השנייה, נריץ אותה באופן היפותטי עם אסטרטגיית השערוך NORMAL, ועל התוכנית להדפיס את המילה NORMAL.
3. בפעם השלישית, נריץ אותה באופן היפותטי עם אסטרטגיית השערוך LAZY, ועל התוכנית להדפיס את המילה LAZY.

```
#include <stdio.h>
int X = 0;
int Y = 0;
int func1(int x){
X++;
return 1;}
int func2(int x){return x + x;}
int func3(int x){
Y++;
return x;}
void main(){
int x = func2(func1(func3(1)));
if(X==2){ //normal, func1 will be evaluated 2 times.
printf("NORMAL");}
else if(X==1){
	if(Y==1){//eager, func1 will be evaluated one time and func3 is evaluated once.
		printf("EAGER");}
	else if(Y==0){//lazy, func1 will be evaluated one time and there is no need to evaluate
func3.
		printf("LAZY");}
}
}
```

שאלה 2

ענו על הסעיפים הבאים בהקשר של שפת D:

1. מהי ברירת המחדל עבור שיטת השערוך בשפה? האם המתכנת יכול להשפיע על שיטת השערוך? אם כן, איך?

ברירת המחדל עבור שיטת השערוך היא EAGER , אך ביכולתו של המתכנת להשפיע על שיטת השערוך על מנת להפוך אותה לLAZY על ידי שימוש ב-
delegate parameter,global parmaters דוג לשימוש זה:

```
void log(const(char)[] delegate() dg)
{
   if (logging)//global parameter
      fwritefln(logfile, dg());
}

void foo(int i)
{//string evaluate only if logging is true
   log( { return "Entering foo() with i set to " ~ toString(i);
});

}
```

2. האם קיימות בשפה פונקציות אנונימיות? אם כן, תנו דוגמה.

```
int function(char c) fp;

void main()
{
  static int foo(char c) { return 6; }
  fp = &foo;
}
```

3. האם קיימים בשפה generators? אם כן, כתבו generator בשפת D שמקבל שני מספרים a, b ומחזיר את מספרי סדרת פיבונאצ'י שמתחילה מ-a, b.

```
auto fib = new Generator!int( { yield(a); yield(b); while(true){ int temp = b; b = a+b; a = temp;
yield(b); } })
```

שאלה 3

ענו על שאלת המבחן הבאה: שאלה 2 מאביב 2019 מועד ב (שלושת הסעיפים).

א. הסבירו את המושגים:

NORMAL ORDER EVALUATION:

שערוך נורמלי הוא שיטות שיערוך דומה לשערוך עצל(הפרמטרים משוערכים בזמן השימוש הראשון בהם על ידי הפונקציה הנקראת) אך במקרה זה הפרמטר משוערך בכל פעם שיש בו שימוש מחדש (כלומר, ערכו לא ישמר לשימושים הבאים בפונקציה).

LEXICAL SCOPING :

ב-LEXICAL SCOPING ה-BINDINGS שלא נוצרו בפונקציה נקבעים לפי ה**מקום** שבו הפונקציה **מוגדרת**.

פונקציה "יורשת" את ה-BINDINGS של פונקציה אחרת רק אם היא מוגדרת בתוכה. הסביבה מוגדרת סטטית ונקבעת על ידי ה-SCOPE.

LEXICAL SCOPING נקרא לעיתים STATIC SCOPING.

ב. הסבירו את השינויים שיש לעשות במימוש של MLISP על מנת לתמוך ב:

NORMAL ORDER EVALUATION:

על מנת לשנות את שיטת השערוך עלינו לבצע את השינוי הבא: כאשר מבצעים את תהליך הBINDING בין הפרמטרים האקטואליים לפרמטרים הפורמליים, לא נבצע שערוך של הפרמטרים הממשיים אלא ניצור כריכה בין הפרמטר הפורמלי לביטוי המייצג את הפרמטר האקטואלי מבלי לשערך אותו. יתרה מכך, כאשר נעשה שימוש בפרמטר בגוף הפונקציה נשערך אותו מחדש נקרא לeval עם הכריכה בה הגדרנו אותו.

LEXICAL SCOPING :

נשמור כל פונקציה כזוג סדור שבו האיבר הראשון הוא הפונקציה והאיבר השני הוא מצביע לסביבה ממנה נקראת הפונציה. בכל פעם שנקרא לפונקציה יש לשמור מצביע לסביבה הנוכחית ואז לעבור לסביבה של הפונקציה, לבצע אותה, ובסיום לחזור אל הסביבה ממנה הפונקציה נקראה דרך המצביע ששמרנו. בדומה לשימוש ברגיסטר ra.

ג.הרחבת הקוד:

```
CONS(
ATOM(SYMBOL("if")),
        CONS(
        CONS(boolean_expression, ATOM(NIL)),
        CONS(
                CONS(true_expression, ATOM(NIL)),
        CONS(CONS(false_expression, ATOM(NIL)), ATOM(NIL))
                )
        )
) => if((eval boolean_expression env)==ATOM(SYMBOL("true"))) then
        (eval true_expression env)
    else if((eval boolean_expression env)==ATOM(SYMBOL("false"))) then
        (eval false_expression env)
    else raise (NOT_BOOLEAN_EXPRESSION);
```