

## שאלה 1 (33 נקודות)

יש לענות על הסעיפים הבאים על פי הקוד הנתון:

```
def f(x,y):  
    return (x+y) % 2  
  
def g(x):  
    if len(x) == 1:  
        return x[0]  
    else:  
        return x[0] + g(x[1:])
```

א. (3 נקודות)

```
print(g([0,1,0,1,0]))    2  
print(g([1,1,1,1,0]))    4
```

מה יודפס?

ב. (1 נקודה)

הסבירו במילים מה מחשבת הפונקציה g. **סכום האיברים ברשימת הקלט**

```
def h(x):  
    if len(x) == 1:  
        return x[0]  
    else:  
        return f(x[0],h(x[1:]))
```

ג. (4 נקודות)

```
print(h([0,1,0,1,0]))    0  
print(h([1,1,1,1,0]))    0
```

מה יודפס?

```
def foo(x,y):  
    return g([f(x[i],y[i]) for i in range(len(y))])
```

ד. (3 נקודות)

```
print(foo([0,0,1,1,1,0,0], [1,1,1,1,1]))
```

2

מה יודפס?

ה. (2 נקודות)

הסבירו במילים מה מחשבת הפונקציה foo (מספיק להסביר מה יחושב עבור ארגומנטים שהם רשימות המכילות 0,1 בלבד).

**עבור שתי רשימות של 0-1 שהראשונה לא יותר קצרה מהשנייה:  
מספר המקומות שבהם הרשימות שונות, עד אורך הרשימה השנייה.**

```
def bar(x,y,z):
    if len(x) < len(y):
        return False
    elif foo(x,y) <= z:
        return True
    else:
        return bar(x[1:],y,z)
```

ו. (6 נקודות)

```
print(bar([0,0,1,1,1,0,0,0,1,1,0,0,1,0,0,0], [1,1,1,1,1], 1)) False
print(bar([1,0,1,0,0,0,0,0,1,1,0,0,1,0,0,0], [1,1,1,1,1], 2)) True
```

מה יודפס?

ז. (4 נקודות)

רשמו את הפרמטרים (x,y,z) של הקריאה הרקורסיבית האחרונה (הקריאה העמוקה ביותר ברקורסיה) ל-  
bar בשורה הראשונה שבסעיף ו. `bar([1,0,0,0], [1,1,1,1,1], 1)`

ח. (4 נקודות)

שנו את הקוד של bar כך שהפונקציה תעשה אותו הדבר אבל ללא רקורסיה. **הקוד מצורף בנפרד**

ט. (3 נקודות)

ציינו לפחות שיפור אחד בסיבוכיות הזמן או המקום שהמימוש בסעיף הקודם נותן על פני המימוש המקורי (מספיק לציין את סיבוכיות המקום או הזמן בסעיפים הנ"ל, ולנמק במילים ללא הוכחה מתימטית של הסיבוכיות) **ההסבר מצורף בנפרד**

```
def foo_foo(x,y,z):
    if bar(x,y,z):
        for i in range(len(x)):
            if len(x) >= len(y)+i and foo(x[i:],y) <= z:
                yield i
```

י. (3 נקודות)

```
for i in foo_foo([0,1,0,1,0,0,1,1,1,1,0,1,0,1,1,1,1], [1,1,1,1,1], 1):
    print(i)
```

מה יודפס?

5  
6  
7  
11  
12

## סעיף ח':

```
def bar(x,y,z):  
    for i in range(len(x)):  
        if len(x[i:]) < len(y):  
            return False  
        elif foo(x[i:],y) <= z:  
            return True
```

## סעיף ט':

נסמן ב- $n$  את האורך של  $x$  וב- $m$  את האורך של  $y$ .  
הרקורסיה היא בעומק של  $n-m+1$ . אם  $n$  משמעותית גדול מ- $m$  זה סדר גודל של  $n$ .  
סיבוכיות המקום היא סדר גודל של  $n^2$  - הרמה ה- $i$  ברקורסיה מייצרת מחרוזת באורך  $n-i$  וכל המחרוזות הללו נשמרות במקביל בזיכרון.  
לעומת זאת, במימוש הלא-רקורסיבי, שומרים בכל זמן נתון רק מחרוזת אחת באורך של לכל היותר  $n$ .  
במימוש מתוחכם יותר, אפשר לחסוך גם בזמן ולא רק במקום.