

שפות תכנות -- מבחן מועד א', סמסטר א', 2022/2023

- יש לענות אך ורק בטופס המבחן. מחברת הבחינה לא תיבדק.
- המחלקה למדעי המחשב
- מרצה: ד"ר יוני זוהר
- מתרגל: צביקה ברגר
- חומר עזר חיצוני: אין
- משך המבחן: 180 דקות
- מומלץ להשתמש במחברת הבחינה כדי לכתוב טיוטה של התשובות, ואז להעתיקן למקום המתאים בטופס.
- במידת הצורך, בסוף הטופס ישנם דפים נוספים בהם ניתן לכתוב תשובות לשאלות.
- בהצלחה!
- שוב: יש לענות אך ורק בטופס המבחן. מחברת הבחינה לא תיבדק.

1. להלן ריצה אינטראקטיבית של `utop`, האינטרפרטר של `OCaml`. שורה שמתחילה ב"`utop #`" היא שורת קלט מהמשתמש. שורה ללא `utop #` היא התשובה של `utop`. מהריצה הושמטו חלקים, המסומנים בקו. השלימו אותם. אם מתקבלת שגיאה, אין צורך לרשום אותה במדויק, ומספיק לרשום `.error`.

(א)

```
utop # let curry = (fun f -> (fun x -> (fun y -> (f (x, y)))));;
```

```
val curry : _____ = <fun>
```

(ב)

```
utop # let f = fun (x,y) -> x+y;;
```

(ג)

```
utop # (curry(f))(3,4);;
```

(ד)

```
utop # (curry(f))(4)(3);;
```

(ה)

```
utop # (curry(f))(4);;
```

2. להלן קטע קוד באוקמל שחלקו הושמט.

```
let rec fold_right = fun f l init -> match l with  
  [] -> init  
  | h::t -> f h (fold_right f t init);;
```

```
let f = _____;;
```

```
let init = _____;;
```

```
let product = fun ell -> fold_right f ell init;;
```

(א) השלימו את ההגדרות של f ו- $init$ כך ש- $product$ תהיה פונקציה שמקבלת רשימה ומחזירה את מכפלת כל איבריה. אין זה משנה מה $product$ תחזיר עבור הרשימה הריקה.

(ב) מה הטיפוס של $fold_right$?

3. להלן הגדרות ב-OCaml (תזכורת: @ הוא אופרטור השרשור בין רשימות):

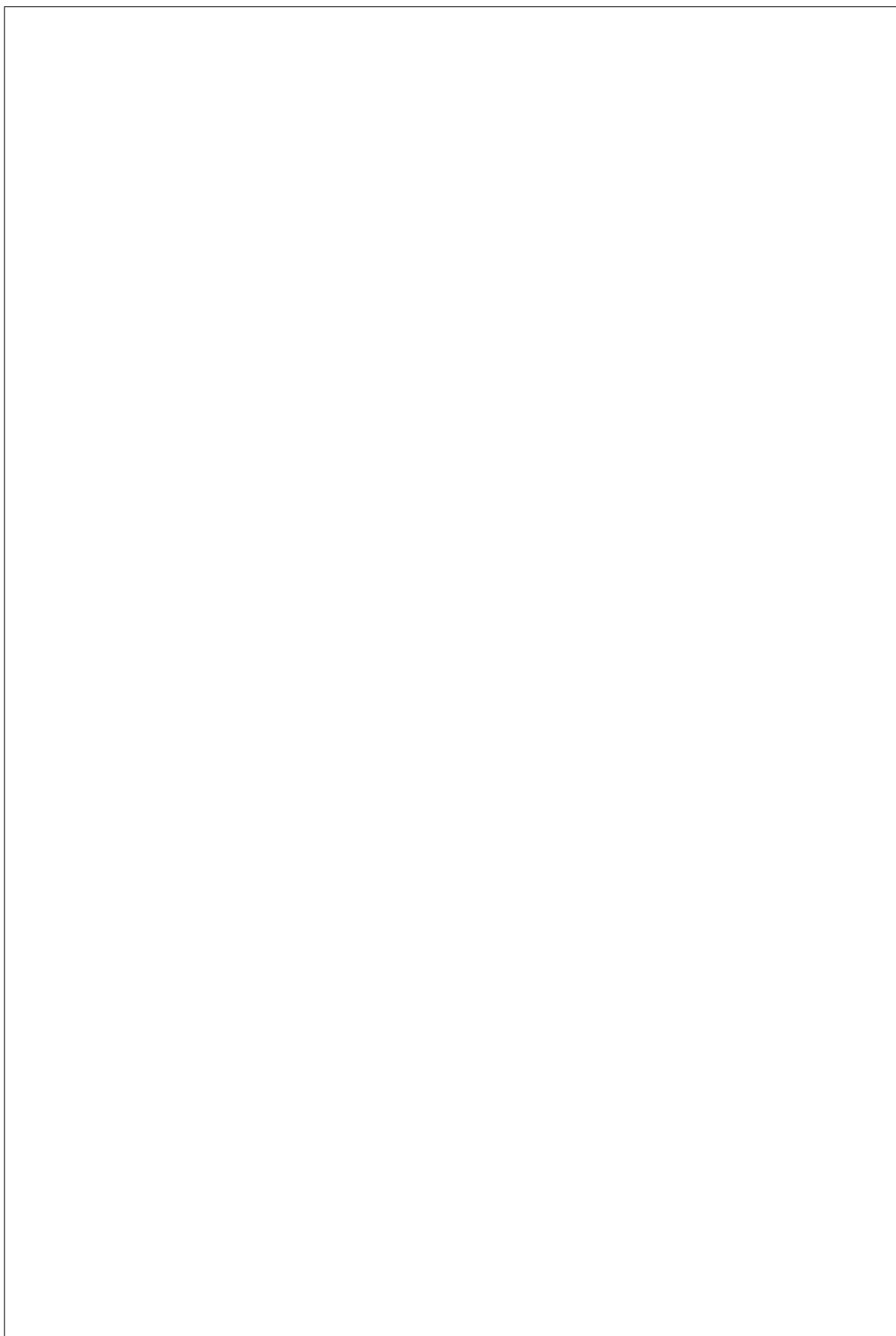
```
let rec len = fun l -> match l with [] -> 0 | h::t -> 1 + len(t);;  
let rec rev = fun l -> match l with [] -> [] | h::t -> (rev t) @ [h];;
```

הוכיחו באינדוקציה כי $len(rev(\ell)) = len(\ell)$.

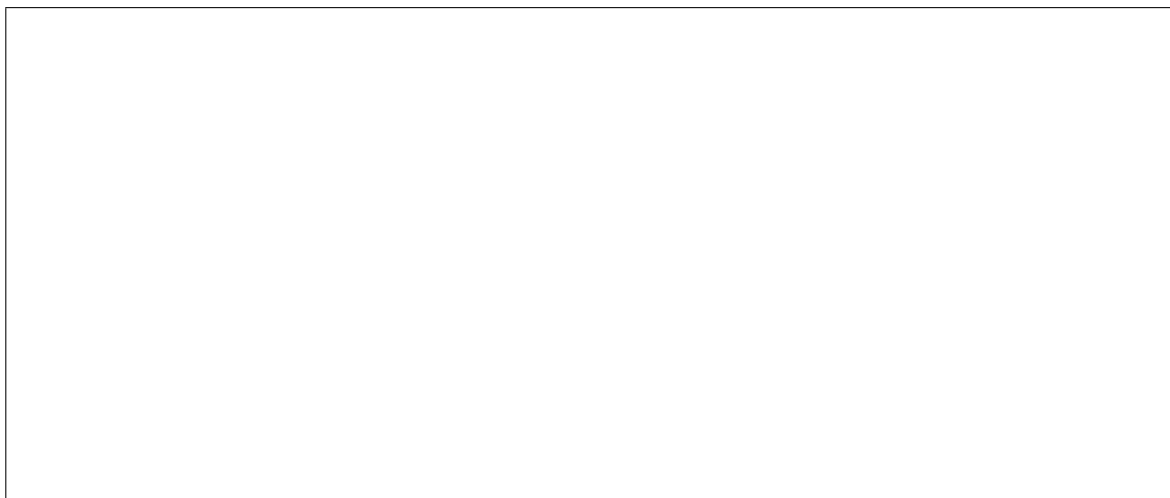
תוכלו להשתמש בלמה הבאה מבלי להוכיח אותה: $len(\ell_1 @ \ell_2) = len(\ell_1) + len(\ell_2)$.



4. יהי s מצב כך ש- $5 \leq s < 6$. הוכיחו כי: $\langle \text{while } x \leq y \text{ do } x := x + 1, s \rangle \rightarrow s [x = 7]$ על ידי שימוש בכללי ההיסק של הסמנטיקה הטבעית.

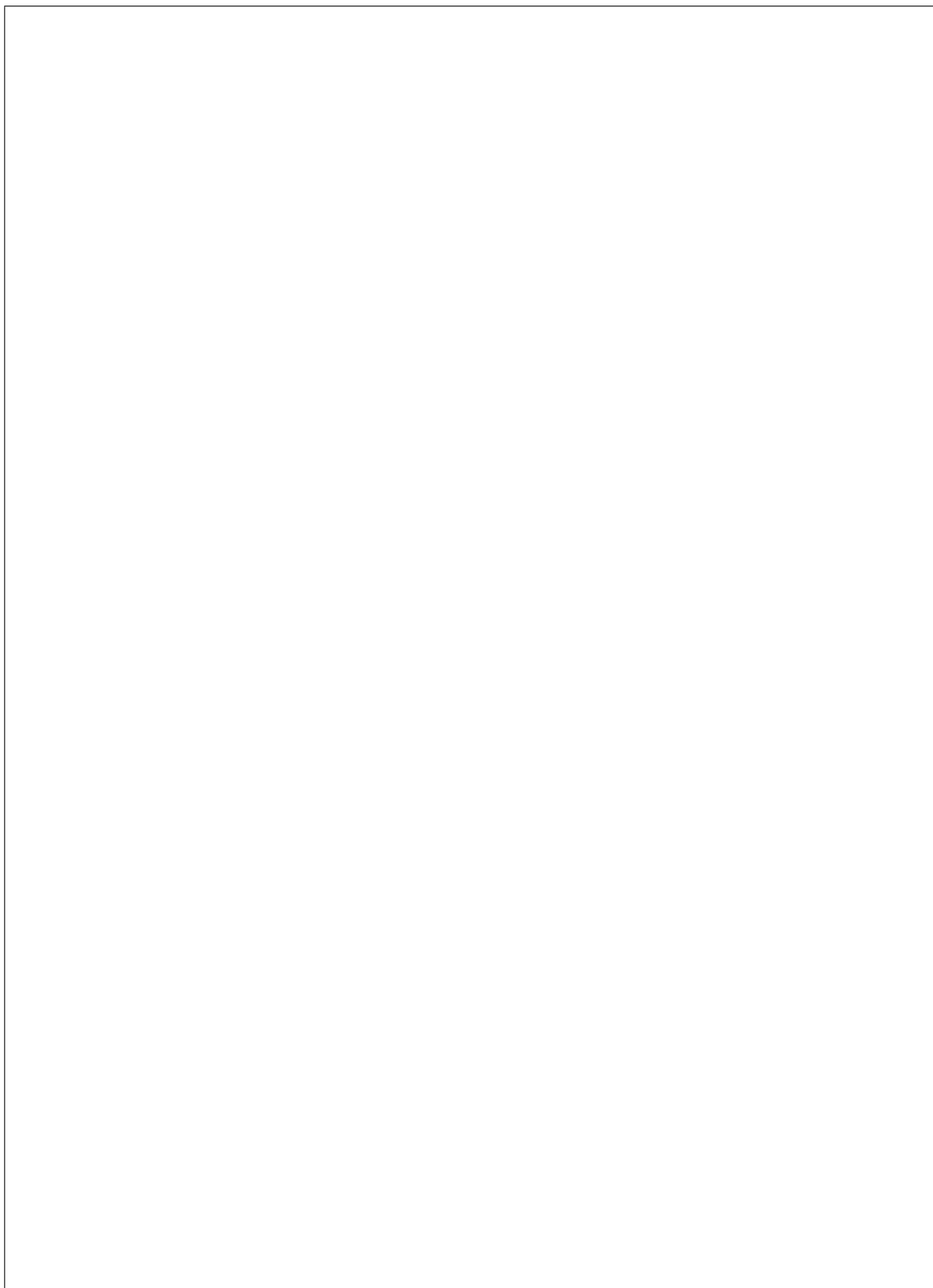


5. הגדירו קומבינטור $isodd$ שיקיים לכל $m \geq 0$: אם m זוגי אז $isodd c_m = fls$ ואם m אי-זוגי אז $isodd c_m = tru$. מותר להשתמש בקומבינטורים מדף הנוסחאות מבלי להגדירם מחדש. כל קומבינטור אחר שתמצאו להשתמש בו, יש להגדיר במפורש.



6. הפעילו רדוקציות β על הביטוי הבא עד שתגיעו לביטוי שממנו לא ניתן לעשות רדוקציית β . בצעו את הרדוקציות באיזה סדר שתבחרו.

$$scc\ c_0$$



(א) הוכיחו על ידי שימוש בכללי ההטפסה: $\vdash \lambda x : Bool. \text{if } x \text{ then true else } x : Bool \rightarrow Bool$

(ב) מצאו T כך ש- $\vdash (f x) y : Bool$ $x : Bool, y : Bool, f : T$

8. להלן ביטוי בשפת אוקמל:

```
let x = 1 in
  let f = fun y -> x+6 in
    let x = 2 in
      f 1;;
```

(א) חשבו את ערכו תחת *static – scoping*

(ב) חשבו את ערכו תחת *dynamic – scoping*

(ג) להלן חלק מגזירה של אבלואציה של ביטוי. קבעו מי הם $?_1, ?_2, ?_3$.

$$\frac{\frac{\frac{}{\Box \vdash 1 \Rightarrow ?_2} \text{Constant}}{(x : 1) :: \Box \vdash \text{fun } y \rightarrow x + 6 \Rightarrow ?_1} \text{Fun1} \quad \vdots}{(x : 1) :: \Box \vdash (\text{let } f = \text{fun } y \rightarrow x + 6 \text{ in } (\text{let } x = 2 \text{ in } (f 1))) \Rightarrow 1} \text{Let}}{\Box \vdash \text{let } x = 1 \text{ in } (\text{let } f = \text{fun } y \rightarrow x + 6 \text{ in } (\text{let } x = 2 \text{ in } (f 1))) \Rightarrow ?_3} \text{Let}$$

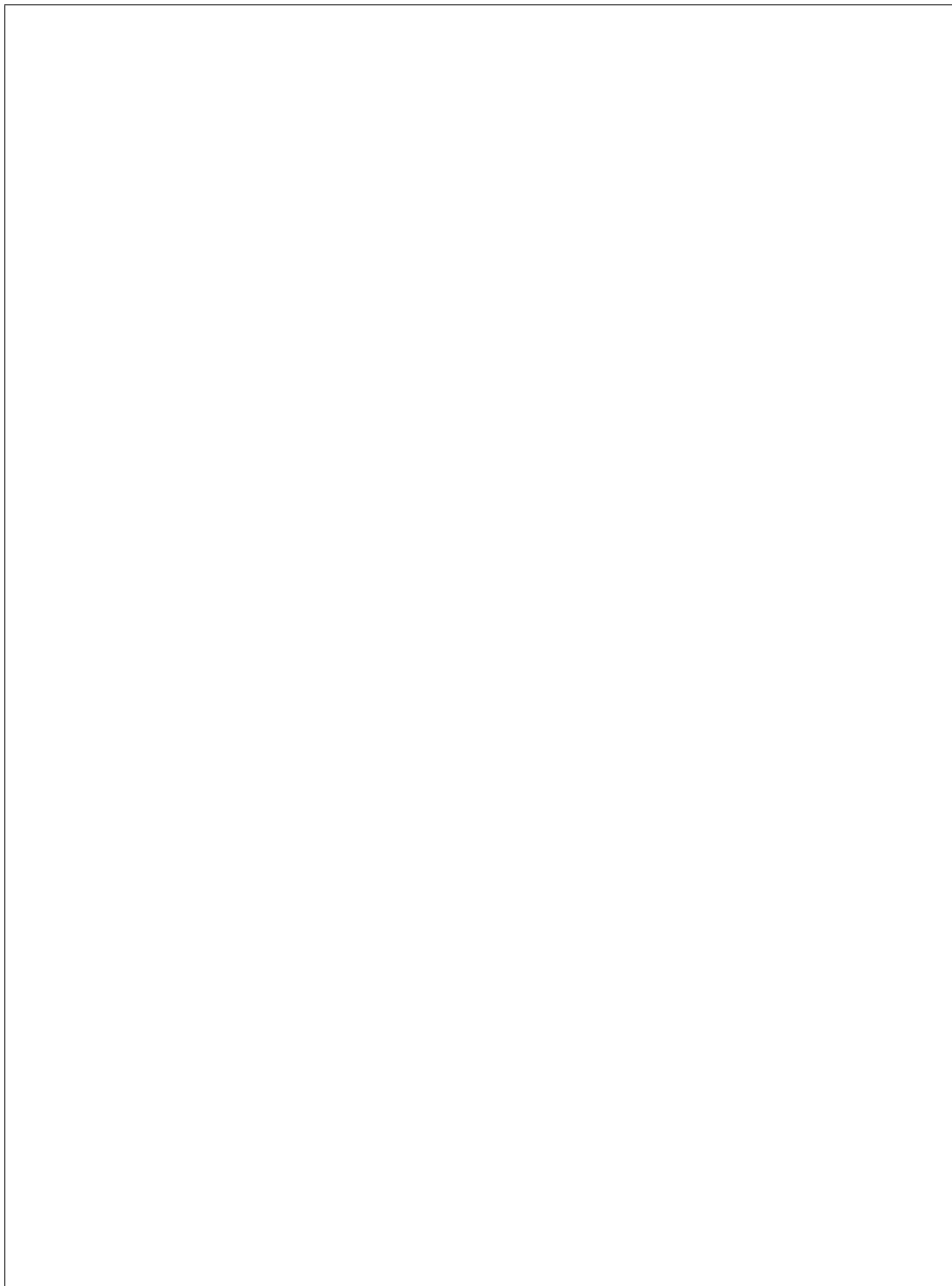
• $?_1$ הוא

• $?_2$ הוא

• $?_3$ הוא

9. כתבו תכנית פרולוג בשם *my_last* שמחזירה את האיבר האחרון ברשימה. אין צורך להתייחס למקרה בו הרשימה ריקה, ואין צורך להשתמש במילה *assert*. להלן ריצה לדוגמה:

?- *my_last*(X, [a, b, c, d]).
X = d



10. להלן קוד בג'אווה סקריפט עם חלקים שהושמטו, שמטרתו היא לממש את פונקציית עצרת עם *memoization*. לכל $n \geq 1$ עליה להחזיר את $n!$ השלימו את המקומות החסרים. ניתן לכתוב מספר פקודות באותה השורה, אך במקרה כזה יש להפרידן ב-
;

```
let cache = [];
```

```
cache[1] = _____
```

```
function memoizeFactorial(n) {
```

```
  if (cache[n]) {
```

```
    _____
```

```
  } else {
```

```
    _____
```

```
  }
```

```
}
```



