

<u>שאלה 3 (25 נקודות)</u>

עבור שני מספרים שלמים a ו-b, כאשר b>a, נגדיר **פרמוטציה 1-חסרה** של התחום [a,b] כמערך המכיל את כל השלמים בין a ל-b, כל מספר בדיוק פעם אחת, פרט לאחד מהם שחסר מהמערך. אנו נאמר שפרמוטציה 1-חסרה היא **ממוינת** אם האיברים בה מסודרים בסדר עולה.

לדוגמה,

- המערך {4,3,1,7,6,5} הוא פרמוטציה 1-חסרה של התחום [1,7], כיוון שהאיבר 2 חסר.
- 2. המערך (2,4,5,6) הוא פרמוטציה 1-חסרה ממוינת של התחום [2,6], כיוון שהאיבר 3 חסר.

<u>סעיף א</u>

ממשו את הפונקציה הבאה, המקבלת כקלט מערך p המכיל פרמוטציה 1-חסרה **ממוינת** של התחום [a,b], ומחזירה את הערך החסר במערך. **על מימוש הפונקציה להיות רקורסיבי**; פתרון לא רקורסיבי יזכה לניקוד חלקי בלבד.

<u>דרישות סיבוכיות</u>: O(log(n)) זמן, O(log(n)) מקום נוסף, כאשר n הוא אורך המערך p. שימו לב: פתרון בסיבוכיות גרועה מזו יקבל ניקוד חלקי בלבד.

<pre>int findsorted(int p[], int a, int b) {</pre>



<u>סעיף ב</u>

בסעיף זה ניתן להשתמש בפונקצית העזר הבאה:

```
int pivot(int a[], int n, int key);
```

פונקציה זו מקבלת מערך a באורך n וערך key, ומשנה את סדר אברי המערך כך שכל האברים הקטנים או key, שווים ל-key נמצאים בתחילת המערך, וכל האברים הגדולים ממנו נמצאים בסוף המערך. הפונקציה מחזירה את האינדקס שהחל ממנו ממוקמים האברים הגדולים מ-key במערך. דהיינו, כל האיברים שקטנים או שווים את האינדקס שהחל ממנו ממוקמים האברים הגדולים מ-key נמצאים ב-[n-1],...,a[index],...,a[n-1].

הפונקציה ()pivot פועלת בסיבוכיות זמן (O(n וסיבוכיות מקום נוסף (O(1).

ממשו את הפונקציה הבאה, המקבלת כקלט מערך p המכיל פרמוטציה 1-חסרה **כלשה**י של התחום [a,b], ומחזירה את הערך החסר במערך. **על מימוש הפונקציה להיות רקורסיב**י; פתרון לא רקורסיבי יזכה בניקוד חלקי בלבד. שימו לב ש-p איננו בהכרח ממוין בסעיף זה.

<u>דרישות סיבוכיות</u>: O(n) זמן, O(log(n)) מקום נוסף, כאשר n הוא אורך המערך p. שימו לב: פתרון בסיבוכיות גרועה מזו יקבל ניקוד חלקי בלבד.

<pre>int find(int p[], int a, int b) {</pre>