



שאלה 3 (25 נקודות)

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

סעיף א

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.

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

```
int findsorted(int p[], int a, int b)
{
    int mid, n;

    n = b-a;
    if (n==0)
        return a;

    if (n==1) {
        if (p[0]==a)
            return b;
        else
            return a;
    }

    mid = n/2;

    if (p[mid] == a+mid)
        return findsorted(p+mid+1, a+mid+1, b);
    else
        return findsorted(p, a, a+mid);
}
```

הסבר: בכל שלב (קריאה רקורסיבית) נשווה את האיבר האמצעי במערך ($p[mid]$) עם הערך שהיה אמור להיות בו לו כל האיברים מ- a ועד $a+mid$ היו במערך: אם ערכו הוא $a+mid$, סימן שכל האיברים הללו נמצאים ולכן ניתן להמשיך את החיפוש במחצית הימנית של המערך בלבד, ואחרת (דהיינו ערכו של $p[mid]$ הוא $a+mid+1$) האיבר החסר הוא בצד שמאל של המערך, ולכן נמשיך את החיפוש שם. אנו עוצרים כשמגיעים למערך שאורכו 1 או 0. ואז קל לקבוע איזה איבר חסר.



סעיף ב

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 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 ל- key נמצאים בתאים $a[0], \dots, a[index-1]$, ואילו האברים הגדולים מ- key נמצאים ב- $a[index], \dots, a[n-1]$.

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

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

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

```
int find(int p[], int a, int b)
{
    int mid, n, i;

    n = b-a;
    if (n==0)
        return a;

    if (n==1) {
        if (p[0]==a)
            return b;
        else
            return a;
    }

    mid = pivot(p, n, (a+b)/2);
    if (mid == n/2)
        return find(p, a, a+mid);
    else
        return find(p+mid, a+mid, b);
}
```

הסבר: הרעיון כאן דומה לסעיף הקודם. בכל קריאה רקורסיבית נבצע פעולת $pivot$ לאברי המערך, כך שיהיו מסודרים סביב ערך האמצע $(a+b)/2$ – האיברים הקטנים יותר משמאל, והגדולים יותר מימין. כעת ניתן לקבוע באיזו מחצית של המערך נמצא הערך החסר, שהרי mid מקבל בעצם את מספר האיברים במערך שקטנים או שווים ל- $(a+b)/2$: אם כל האיברים בין a ל- $(a+b)/2$ נמצאים במערך ערכו יהיה $n/2+1 = (b-a)/2+1 = (a+b)/2-a+1$ ולכן נמשיך את החיפוש בצד ימין, ואחרת (אם חסר איבר כלשהו בין a ל- $(a+b)/2$) ערכו יהיה $n/2$ בלבד ולכן נמשיך את החיפוש בצד שמאל.