



שאלה 2 (25 נקודות)

סעיף א

נתון מערך a המכיל מספרים שלמים וגדולים ממש מאפס, ממוינים בסדר עולה. עליכם לממש פונקציה המקבלת את המערך a , את גודלו n ומספר שלם חיובי x , ומחזירה 1 במידה וניתן להציג את x כמכפלה של שני מספרים מ- a (שימו לב שמכפלה זו יכולה לכלול שני איברים שונים מ- a , או מכפלת איבר בעצמו). במידה ולא ניתן להציג את x כמכפלה של אברים מ- a , הפונקציה תחזיר 0.

לדוגמה, עבור המערך $a[6] = \{2, 2, 3, 5, 8, 9\}$, הקריאות $\text{two_prod}(a, 6, 18)$ ו- $\text{two_prod}(a, 6, 25)$ יחזירו שתיהן 1 כיוון ש- $2 \cdot 9 = 18$ וכן $5 \cdot 5 = 25$. לעומת זאת, הקריאה $\text{two_prod}(a, 6, 11)$ תחזיר 0 כיוון שאין זוג כלשהו של מספרים במערך שמכפלתם 11.

דרישות סיבוכיות: $O(n)$ זמן, $O(1)$ מקום נוסף. פתרון לא אופטימלי יזכה בניקוד חלקי בלבד.

```
int two_prod(int a[], int n, int x) {
{
    int i=0, j=n-1;

    while (i<=j)
    {
        if (a[i]*a[j] == x)
            return 1;

        if (a[i]*a[j] < x)
            i++;
        else
            j--;
    }
    return 0;
}
```

הסבר: נחזיק שני אינדקסים למערך – שמאלי וימני. בכל איטרציה נזיז אחד מהם, בהתאם להשוואה עם הערך x : אם המכפלה של שני הערכים גדולה מ- x נקטין את האינדקס הימני, ואם המכפלה קטנה מ- x נגדיל את השמאלי. אם בתהליך זה מצאנו זוג אינדקסים כך שהמכפלה שווה בדיוק ל- x – נחזיר 1. לעומת זאת, אם האינדקס הימני עובר את השמאלי אזי בהכרח אין זוג ערכים במערך שמכפלתו שווה ל- x , ולכן נחזיר 0.



סעיף ב

בדומה לסעיף א', נתון מערך a המכיל מספרים שלמים וגדולים ממש מאפס, ממוינים בסדר עולה. עליכם לממש פונקציה המקבלת את המערך a , את גודלו n ומספר שלם כלשהו x , ומחזירה 1 במידה וניתן להציג את x כהפרש של שני איברים מ- a . גם כאן, כמו בסעיף א', ניתן בהפרש זה לחסר שני איברים שונים מ- a , או איבר מעצמו. במידה ולא ניתן להציג את x כהפרש של איברים מ- a , הפונקציה תחזיר 0.

לדוגמה, עבור המערך $a[6] = \{2, 2, 3, 5, 8, 9\}$, הקריאות $\text{two_diff}(a, 6, 7)$ ו- $\text{two_diff}(a, 6, -3)$ יחזירו שתיהן 1 כיוון ש- $9-2=7$ וכן $-3=5-8$. לעומת זאת, הקריאה $\text{two_diff}(a, 6, 8)$ תחזיר 0 כיוון שאין זוג כלשהו של מספרים במערך שהפרשם 8.

דרישות סיבוכיות: $O(n)$ זמן, $O(1)$ מקום נוסף. פתרון לא אופטימאלי יזכה בניקוד חלקי בלבד.

```
int two_diff(int a[], int n, int x)
{
    int i=n-1, j=n-1;

    while (i>=0 && j>=0)
    {
        if (a[i]-a[j] == x)
            return 1;

        if (a[i]-a[j] < x)
            j--;
        else
            i--;
    }

    return 0;
}
```

הסבר: בדומה לסעיף הקודם, נחזיק שני אינדקסים למערך, אולם הפעם שניהם מאותחלים לצד ימין של המערך. בכל איטרציה נקטין אחד מהם על פי תוצאת ההשוואה עם x : אם ההפרש גדול מ- x נקטין את i , ואחרת נקטין את j . אם בתהליך זה מצאנו זוג אינדקסים שהפרשם שווה ל- x נחזיר 1, ולעומת זאת אם אחד מהם מגיע עד לתחילת המערך בלא שמצאנו זוג מתאים, נחזיר 0.