



שאלה 1 (20 נקודות)

סעיף א (10 נקודות)

בשאלה זו נתייחס לאותיות האנגליות הקטנות $a \dots z$ כספרות, כאשר 'a' ערכה 0, 'b' ערכה 1, 'c' ערכה 2 וכן הלאה עד 'z' שערכה 25. כעת, אותיות אלו יישמשו אותנו לצורך ייצוג מספרים ממשיים חיוביים (גדולים או שווים ל-0) בבסיס 26. כל מספר ייוצג כמחרוזת בפורמט הבא:

$$a_{k-1}a_{k-2}\dots a_1a_0 \cdot a_{-1}a_{-2}\dots a_{-p}$$

כאשר כל a_i מייצג אות אנגלית קטנה. במספר זה ישנם k "ספרות" לפני הנקודה ועוד p "ספרות" אחרי הנקודה, ולצורך השאלה ניתן להניח כי $k \geq 1$ וגם $p \geq 1$. ערכו של המספר בייצוג זה נתון על ידי הביטוי:

$$a_{k-1} \cdot 26^{k-1} + a_{k-2} \cdot 26^{k-2} + \dots + a_0 \cdot 26^0 + a_{-1} \cdot 26^{-1} + \dots + a_{-p} \cdot 26^{-p}$$

דוגמאות:

- ערכו של הביטוי "a.a" הינו 0.
- ערכו של הביטוי "ba.c" הינו $1 \cdot 26^1 + 2 \cdot 26^{-1} = 26.077$

השלימו את הפונקציה הבאה, המקבלת כקלט מחרוזת s המייצגת מספר חיובי בבסיס 26, ומחזירה את ערכו של המספר כ-`double`. ניתן להניח כי s מכילה קלט חוקי. יש לבצע את החישוב במעבר יחיד על המחרוזת.

```
double strval(char *s) {  
  
    double base = 1, result = 0;  
  
    while (*s != '.') {  
        result *= 26;  
        result += (*s - 'a');  
        ++s;  
    }  
  
    while (*(++s) != 0) {  
        base /= 26;  
        result += (*s - 'a') * base;  
    }  
  
    return result;  
}
```



סעיף ב (10 נקודות)

נתונים שני מערכים a ו- b . כל אחד מהם הינו מערך של מחרוזות, כאשר כל מחרוזת מייצגת מספר בבסיס 26 כמתואר בסעיף א'. שני המערכים **ממוינים** בסדר עולה, לפי הערך המספרי של המחרוזות שהם מכילים.

כתבו פונקציה המקבלת שני מערכים ממוינים a ו- b כנ"ל, בגודל n , m בהתאמה, ומספר ממשי x . הפונקציה מחזירה 1 אם קיימים זוג איברים $a[i]$ ו- $b[j]$ כך ש- $a[i] - b[j] == x$ (שימו לב שהכוונה רק לאיבר מהמערך a פחות איבר מהמערך b , ולא להיפך). במידה ולא קיימים שני איברים כאלו הפונקציה תחזיר 0.

בשאלה זו נניח כי כל המחרוזות במערכים a ו- b הינן באורך לכל היותר c , כאשר c הוא קבוע ידוע. תחת הנחה זו, על הפונקציה שאתם כותבים בסעיף זה לעבוד בסיבוכיות זמן $O(n+m)$, וסיבוכיות מקום נוסף $O(1)$.

```
int finddiff(char* a[], int n, char* b[], int m, double x) {
```

```
    int i = 0, j = 0;
```

```
    double diff;
```

```
    while (i < n && j < m) {
```

```
        diff = strval(a[i]) - strval(b[j]);
```

```
        if (diff == x) return 1;
```

```
        if (diff < x) i++;
```

```
        else j++;
```

```
    }
```

```
    return 0;
```

```
}
```