



## שאלה 2 (20 נקודות)

נתונה מחרוזת המורכבת מאותיות אנגליות קטנות בלבד. פרמוטציה של מחרוזת זו מוגדרת כמחרוזת המכילה את אותו תוכן בדיוק, אולם ייתכן שבשינוי סדר (שימו לב שכל מחרוזת הינה פרמוטציה של עצמה).

לדוגמה,

- עבור המחרוזת "abc", פרמוטציות אפשריות הן "abc", "cab", "acb" (ויש פרמוטציות נוספות).
- עבור המחרוזת "abba", פרמוטציות אפשריות הן "abba", "aabb", "baba" וכן הלאה.

השלימו את הפונקציה הבאה, המקבלת שתי מחרוזות  $s_1$  ו- $s_2$ , ומחליטה האם הן פרמוטציה אחת של השנייה. הפונקציה מחזירה 1 אם כן, ו-0 אם לא. ניתן להניח ששתי המחרוזות חוקיות, ומכילות אותיות אנגליות קטנות בלבד (a-z).

על הפונקציה לעבוד בסיבוכיות זמן  $O(n+m)$  (כאשר  $n$  ו- $m$  הינם אורכי שתי המחרוזות בהתאמה), וסיבוכיות מקום נוסף  $O(1)$ .

הערה: ניתן להתייחס למספר האותיות בא"ב האנגלי כאל קבוע.

```
int permute(char* s1, char* s2) {
```

```
    int count[26] = {0}, i;
```

```
    while (*s1) {
```

```
        count[*s1 - 'a']++;
```

```
        s1++;
```

```
    }
```

```
    while (*s2) {
```

```
        count[*s2 - 'a']--;
```

```
        s2++;
```

```
    }
```

```
    for (i=0 ; i<26 ; ++i)
```

```
        if (count[i] != 0) return 0;
```

```
    return 1;
```

```
}
```