

תרגיל בית 4

המתרגלת האחראית על תרגיל זה: קטריין חדאד-זקנון

**שאלות בנוגע לתרגיל בית זה, יש להעביר דרך ה- piazza בלבד.**

הנחיות כלליות:

- הגשה **בבודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרון.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל, עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל לפני תאריך הגשת התרגיל.
- ערעורים ניתן להגיש עד שבוע לאחר קבלת הציון.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה בחלקה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
  - השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
  - וודאו את האותיות הגדולות והקטנות לפי הדוגמאות וההסברים בתרגיל.
  - אין להדפיס רווחים שלא התבקשתם להדפיס (בתחילת שורה או בסופה).
  - בכל סוף שורה יש להדפיס תו ירידת שורה, גם בשורה האחרונה.
  - השתמשו באתר הבדיקה העצמית.
- בתרגיל זה מותר להשתמש בפונקציות מהספרייה `string.h`, `stdio.h`, `stdbool.h` בלבד, שנלמדו בהרצאות ובתרגולים, אלא אם צוין אחרת.
- ההגשה הינה אלקטרונית ובבודדים דרך אתר הקורס. קובץ ההגשה יהיה מסוג zip (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
  - קובץ `students.txt` עם מספר תעודת הזהות שלך וכתובת האי-מייל שלך.
  - קובץ פתרון `hw4q1.c` לשאלה 1
  - קובץ פתרון `hw4q2.c` לשאלה 2
  - קובץ פתרון `hw4q3.c` לשאלה 3
- **חובה לשמור את קוד אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה לא תתקבל ע"י המערכת! אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

הוראות הגשה:

- לתרגיל זה מצורפים קבצי שלד שהם תואמים את התבנית שתקבלו בבחינת מועד א. רצוי שתעיינו במבנה שלהם כדי להתרגל אליהם לפני הבחינה.
- קבצי השלד הם: `hw4q1.c`, `hw4q2.c`, `hw4q3.c` המתאימים לשלושת השאלות של התרגיל.
- קובץ השלד מכיל תבנית של הפתרון עם חתימות של הפונקציות שאותם אתם צריכים לממש בכל שאלה.
- יש להעתיק את תוכן קובץ השלד לסביבת הפיתוח שלכם ולממש את הפונקציות הנדרשות במקום המתאים.

- כל אחד מהקבצים מכיל פונקציית main ופונקציית test. **אין לשנות את התוכן של פונקציות אלו או למחוק אותן.** כמו כן, **אין לשנות את החתימות של הפונקציות הנקראות מתוך הפונקציות main ו-test**, בפרט, אין לשנות את החתימות של הפונקציות שאתם אמורים לממש בתרגיל. במידה ומסופקות לכם פונקציות אחרות מלבד הפונקציות הנ"ל (main ו-test) **אסור לשנות את המימוש שלהן או את חתימתן.**
- בקבצי השלד תמצאו גם את הפונקציות: PrintIDontKnow()/PrintIKnow(). פונקציות אלו אינן קשורות לתרגיל בית זה, אבל הם ישחקו תפקיד בבחינה. השארנו אותם כחלק מהתרגיל, רק כדי שתכירו אותם לקראת הבחינה. נא לא לשנות את הפונקציות ולא למחוק אותן מהקובץ.
- מותר ורצוי להוסיף פונקציות עזר כרצונכם, כמו כן מותר לכם להוסיף הגדרות של קבועים במידת הצורך.
- הנחיות הגשה **נוספות לשאלה מס' 3** תמצאו בגוף השאלה עצמה.

נושא התרגיל:  
מחרוזות, הקצאה דינמית וסיבוכיות

## שאלה 1:

הנחיה כללית:

- בכל אחת מהפונקציות, **אסור לשנות את תוכן המחרוזת**.
- בשאלה זו, קריאת הקלט של התוכנית והפעלת הפונקציות אותם אתם מתבקשים לממש ממומשת בפונקצית `test()`.
- בשאלה זו מותר לכם להניח כי קלט התכנית שלכם מורכב מאותיות אנגליות קטנות בלבד.
- בכל אחת מהפונקציות, מותר לכם להניח כי המחרוזת איננה ריקה – כלומר מכילה לפחות תו אחד.

**הגדרה 1:** מחרוזת `str` תיקרא **פלינדרום** אם קריאתה משמאל לימין ומימין לשמאל היא זהה.  
דוגמא:

המחרוזות הבאות הינם פלינדרומים:

- `Str="aaaaaa"`
- `Str="abcba"`
- `Str=""` (המחרוזת הריקה)
- `Str="abccba"`

**הגדרה 2:** מחרוזת `str` תיקרא **פלינדרום-מעורבב** אם קיים סידור (אחד או יותר) לאותיות שלו כך שהמחרוזת המתקבלת הינה פלינדרום.

דוגמאות:

- `Str="aakkaa"` הינה פלינדרום ולכן בפרט היא פלינדרום-מעורבב
- `Str="aabbc"` הינה פלינדרום מעורבב כיוון שעבור הסידור `Str1="abcba"` המחרוזת `Str1` הינה פלינדרום וכן הסידור `Str2="bacab"` הוא גם פלינדרום המורכב מאותיות המחרוזת `Str`.

**הגדרה 3:** מחרוזת `str` תיקרא **פלינדרום-k-ציקלי** אם המחרוזת המתקבלת מ-`str` לאחר הזזה ציקלית **ימינה** של `k` האותיות האחרונות שלה הינה פלינדרום. הזזה ציקלית ימינה של `k` אותיות אחרונות מוגדרת באופן הבא:

תהי  $str = s_0 s_1 \dots s_{n-1}$ , המילה המתקבלת מהזזה ציקלית ימינה של `k` האותיות האחרונות:

$$str_{new} = s_{n-k} s_{n-k+1} \dots s_{n-1} s_0 s_1 \dots s_{n-k-1}$$

דוגמא:

- עבור המחרוזת `abcdefghijklmn` המחרוזת המתקבלת מהזזה ימינה של 3 אותיות היא `lmnabcdefghijkl`

הערות:

- עבור מחרוזת בגודל `n` אותיות לאחר הזזה של `n` אותיות ימינה נקבל את אותה המחרוזת בחזרה.
- כל פלינדרום-k-ציקלי הוא בפרט פלינדרום-מעורבב (לכל `k`).

המשך בעמוד הבא...

בשאלה זו עליכם לממש את הפונקציות הבאות:

א. עליכם לממש את הפונקציה הבאה:

```
bool isPalindrome(char* str);
```

הפונקציה מקבלת מחרוזת `str` ומחזירה `true` אם המחרוזת הינה פלינדרום ו-`false` אחרת.  
דרישות סיבוכיות:

- **סיבוכיות זמן:**  $O(n)$  כאשר  $n$  הוא אורך המחרוזת כולל סימן `\0`.
- **סיבוכיות מקום נוסף:**  $O(1)$

ב. עליכם לממש את הפונקציה הבאה:

```
bool isMixedPalindrome(char* str);
```

הפונקציה מקבלת מחרוזת `str` ומחזירה `true` אם המחרוזת הינה פלינדרום מעורב ו-`false` אחרת.  
דרישות סיבוכיות:

- **סיבוכיות זמן:**  $O(n)$  כאשר  $n$  הוא אורך המחרוזת כולל סימן `\0`.
- **סיבוכיות מקום נוסף:**  $O(1)$

ג. עליכם לממש את הפונקציה הבאה:

```
bool isKCyclicPalindrome(char* str, int k);
```

הפונקציה מקבלת מחרוזת `str` ומספר שלם אי-שלילי `k` ומחזירה `true` אם המחרוזת הינה פלינדרום `k`-ציקלי ו-`false` אחרת.

דרישות סיבוכיות:

- **סיבוכיות זמן:**  $O(n)$  כאשר  $n$  הוא אורך המחרוזת כולל סימן `\0`.
- **סיבוכיות מקום נוסף:**  $O(1)$

**הערות:** מותר להניח כי `k` הוא אי-שלילי.

ד. עליכם לממש את הפונקציה הבאה:

```
int getLongestMixedSubstring(char* str);
```

הפונקציה מקבלת מחרוזת str ומחזירה את אורך תת-המחרוזת הארוכה ביותר של str שהיא פלינדרום מעורבב.  
דרישות סיבוכיות:

- **סיבוכיות זמן:**  $O(n^2)$  כאשר n הוא אורך המחרוזת כולל סימן \0.
- **סיבוכיות מקום נוסף:**  $O(N)$  כאשר N הוא חסם על אורך המחרוזת str. החסם N מוגדר ע"י define בקובץ השלד המסופק יחד עם התרגיל.

#### דוגמת פלט של בתוכנית:

בקובץ השלד, מסופקת לכם גם פונקצית בדיקה test(). הפונקציה קוראת מהמשתמש מחרוזת str ומספר שלם אי-שלילי k ולאחר מכן מפעילה את ארבעת הפונקציות הנ"ל על הקלט. לפי הערך מוחזר מכל פונקציה, פונקצית הבדיקה מדפיסה למסך הודעה מתאימה. אנא עיינו בקוד של הפונקציה על מנת להבין את פרטי הבדיקה.

דוגמאות הרצה של התכנית:

1

Please enter your string and a non-negative integer...  
abckkkcd 3  
Non-Palindrome  
Non-Mixed Palindrome  
Non-3-Cyclic Palindrome  
Longest mixed substring is:7

1

Please enter your string and a non-negative integer...  
qqqqqqq 5  
Palindrome  
Mixed Palindrome  
5-Cyclic Palindrome  
Longest mixed substring is:7

1

Please enter your string and a non-negative integer...  
baabcddc 2  
Non-Palindrome  
Mixed Palindrome  
2-Cyclic Palindrome  
Longest mixed substring is:8

## שאלה מספר 2

בשאלה זו נתרגל שימוש במחרוזות ובהקצאה דינאמית

- בשאלה זו, קריאת הקלט של התוכנית והפעלת הפונקציות אותם אתם מתבקשים לממש ממומשת בפונקציה `test()`.

### צנזור פוסטים

לקראת תקופת המבחנים הקרבה, הטרולים בקבוצת "סטודנטים בטכניון" מתכננים להפיץ את הקבוצה בפוסטים ויראליים שיגרמו לכולם לבזבז שעות בפייסבוק. כדי שתקופת המבחנים של סטודנטים רבים לא תיהרס, האדמינים של הקבוצה החליטו לצנזר את הפוסטים המתפרסמים בקבוצה, ובפרט למחוק מילים מסוימות אשר עלולות לעורר תגובות סוערות. לשם כך הם פנו אליכם, כך שתכתבו עבורם תכנית הקולטת רשימה של מילים אסורות, ומוחקת אותם מפוסטים. כמו בשאלה הקודמת, רוב התכנית נכתבה עבורכם בקובץ `hw4q2.c`, ועליכם לממש את הפונקציות החסרות.

א. **שחרור זיכרון** : עליכם לממש את הפונקציה:

```
void free_strings(char * strings[], int n);
```

הפונקציה מקבלת מערך של מצביעים `strings` ואת גודלו `n`. על הפונקציה לשחרר את הזיכרון שהוקצה עבור כל אחת מהמחרוזות (ניתן להניח שכל מצביע למחרוזת הוא מצביע שהתקבל מהפונקציה `malloc`). שימו לב – אין לשחרר את הזיכרון של מערך המצביעים עצמו.

ב. **קריאת המילים האסורות**: ממשו את הפונקציה:

```
bool read_strings(char * strings[], int n);
```

הפונקציה מקבלת מערך של מצביעים `strings` ואת גודלו `n`, ועליה לקלוט `n` מחרוזות באמצעות `scanf`. עבור כל מחרוזת יש להקצות זיכרון באמצעות `malloc`, להעתיק את המחרוזת לזיכרון זה, ולשמור את כתובת המחרוזת במערך המצביעים.

ניתן להניח שכל מחרוזת לא חורגת מהאורך `MAX_LEN` אשר מוגדר ב-`define` בתחילת קובץ השלד המסופק עם התרגיל, כאשר אורך זה אינו כולל את התו `'\0'`. שימו לב, בעת הקצאת זיכרון עבור מחרוזת עליכם להקצות את האורך המינימלי הנדרש, כלומר בהתאם לאורכה בפועל, ולא לפי החסם.

במידה ו-`scanf` לא הצליחה לקרוא את אחת המחרוזות או ש-`malloc` נכשל, על הפונקציה להפסיק את קריאת המחרוזות, לשחרר את כל הזיכרון שהקצתה ולהחזיר `false` כערך החזרה שלה. במידה והפונקציה הצליחה, יש להחזיר `true`.

ג. **בדיקת קלט:** ממשו את הפונקציה:

```
bool are_sorted(char * strings[], int n);
```

פונקציה זו מקבלת מערך של מצביעים למחרוזות ואת אורכו, ובודקת האם המערך ממין בסדר לקסיקוגרפי עולה. במידה וכן, היא מחזירה true, ואחרת היא מחזירה false. על המיון להיות case insensitive, כלומר עליו להתעלם מהבדלים בין אותיות גדולות לקטנות, כך שעבור אות מסוימת, קוד ה-ascii שלה אשר לפיו ייבדק המיון, יהיה הקוד של האות הקטנה המתאימה.

דוגמא: עבור רשימת המחרוזות 'cat', 'dig', 'dog', הפונקציה תחזיר true, כי 'cat' קודם ל-'dig' ו-'dig' קודם ל-'dog' בסדר מילוני. מנגד, עבור רשימת המחרוזות 'cat', 'dog', 'dig', הפונקציה תחזיר false.

ד. **בדיקה האם מילה אסורה:** ממשו את הפונקציה:

```
bool is_string_in_array(char * strings[], int n, char * string);
```

פונקציה זו מקבלת מערך ממין בסדר לקסיקוגרפי עולה של מצביעים למחרוזות, את אורכו ומצביע למחרוזת נוספת. על הפונקציה לבדוק האם מחרוזת זו שווה לאחת מן המחרוזות במערך, ולהחזיר true אם כן, ו- false אחרת. על הבדיקה להיות case insensitive, כלומר עליה להתעלם מהבדלים בין אותיות גדולות לקטנות.

לדוגמא, עבור רשימת המחרוזות "Cat", "Dig", "Dog", הפונקציה תחזיר true עבור המחרוזת "dOG" ותחזיר false עבור המחרוזת "Doog".

דרישות סיבוכיות:

- סיבוכיות זמן  $O(MAX\_LEN * \log(n))$ . לצורך הערכת הסיבוכיות, ניתן להניח כי אורכי המחרוזות חסומים ע"י MAX\_LEN.
- סיבוכיות מקום  $O(1)$ .

**חלק ה' – צנזור פוסט:**

ממשו את הפונקציה:

```
void delete_words(char * words[], int n, char * sentence);
```

פונקציה זו מקבלת את מערך המילים words, ממין בסדר לקסיקוגרפי עולה, את אורכו n ומצביע למשפט sentence אשר הינו מחרוזת שבה כל מילה מופרדת באמצעות קו תחתון (התו '\_'). על הפונקציה למחוק מ- sentence את כל המילים אשר מופיעות במערך words כאשר ההשוואה הינה case insensitive.

לדוגמא, עבור רשימת המילים "Cat", "Dig", "Dog" והמשפט "I\_have\_one\_dog\_and\_3\_cats" הפונקציה תשנה את המשפט כך שתוכנו יהיה "I\_have\_one\_and\_3\_cats". באופן דומה, עבור המשפט "I\_love\_my\_dog\_and\_cat", הפונקציה תשנה את המשפט ל-"I\_love\_my\_and".

**דרישות סיבוכיות:**

- עליכם לממש בסיבוכיות זמן וסיבוכיות מקום נוסף קטנות ככל הניתן. כמו מקודם, ניתן להעריך כי אורכי המחרוזות במערך words חסומים ע"י MAX\_LEN, אך אורך המחרוזת sentence אינו

חסום. לצורך חישוב הסיבוכיות, נניח כי מספר המילים ב-sentence הינו לכל היותר  $n$ , כאשר כל מילה במשפט בעלת אורך החסום  $MAX\_LEN$ . אם כן, קריאה לאחת מהפונקציות שלמדנו עליהן בתרגול על כל המחרוזות sentence מתבצעת ב-  $O(MAX\_LEN * n)$ .

#### הערות:

שימו לב, ייתכנו מילים ריקות. לדוגמא במשפט "\_\_\_cat\_" יש 4 מילים, המילה הריקה לפני הקו התחתון הראשון, המילה הריקה בין הקו התחתון הראשון לקו התחתון השני, המילה "cat" והמילה הריקה לאחר הקו התחתון האחרון. לא נסיר מילים ריקות.



### שאלה מס 3:

בשאלה הבאה נתרגל סיבוכיות זמן ומקום של קטע קוד

בשאלות הבאות בחרו בבקשה את האות המייצגת את הסיבוכיות המתאימה מתוך הרשימה הבאה וסמנו את האות המתאימה בקובץ השלד המצורף לשאלה זו, אנא עיינו בהוראות הגשה מפורטות בסוף השאלה:

(a) קטע קוד מס 1: חשבו את סיבוכיות הזמן והמקום של f1 בקטע הקוד הבא:

```
void aux(int n){
    while (n > 0) {
        printf("%d", n);
        n--;
    }
}

long f1(int n)
{
    long sum = 0;
    for (int i = 1; i * i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            aux(i * j);
            sum += i * j;
        }
    }
    return sum;
}
```

1. סיבוכיות הזמן היא: \_\_\_\_\_

2. סיבוכיות המקום היא: \_\_\_\_\_

(b) קטע קוד מס 2: חשבו את סיבוכיות הזמן והמקום של הפונקציה f1

```
void f1(int n)
{
    int x=0;
    for (int i =1; i < n; i*=2)
        for (int j = 0; j < i; j+=2)
            x++;
}
```

3. סיבוכיות הזמן היא: \_\_\_\_\_  
4. סיבוכיות המקום היא: \_\_\_\_\_

(c) קטע קוד מס 3: חשבו את סיבוכיות הזמן והמקום של הפונקציה f2 בקטע הקוד הבא:

```
int f2(int n)
{
    int x =0;
    for (int i =1; i*i <= n; i++)
        for(int j = i; j*j <= n; j+=i)
            free(malloc(++x));

    free(malloc(n));
    return x;
}
```

5. סיבוכיות הזמן היא: \_\_\_\_\_  
6. סיבוכיות המקום היא: \_\_\_\_\_

### הוראות הגשה:

בחרו את האות המייצגת את הסיבוכיות המתאימה מתוך טבלת הפונקציות להלן:

a. $\Theta(1)$ b. $\Theta(\log n)$ c. $\Theta(\log^2 n)$ d. $\Theta(\sqrt{n})$ e. $\Theta(\sqrt{n} \log n)$ f. $\Theta(\sqrt{n} \log^2 n)$ g. $\Theta(n)$ h. $\Theta(n \log n)$ i. $\Theta(n \log^2 n)$ j. $\Theta(n\sqrt{n})$ k. $\Theta(n^2)$ l. $\Theta(n^2 \log n)$ m. $\Theta(n^2 \log^2 n)$	n. $\Theta(n^2 \sqrt{n})$ o. $\Theta(n^3)$ p. $\Theta(n^3 \log n)$ q. $\Theta(n^3 \log^2 n)$ r. $\Theta(2^n)$ s. $\Theta(n * 2^n)$ t. $\Theta(n^2 2^n)$ u. $\Theta(n^3 2^n)$ v. $\Theta(4^n \log n)$ w. $\Theta(4^n \log^2 n)$ x. $\Theta(n 4^n)$ y. $\Theta(n^2 4^n)$ z. <i>the correct answer doesn't appear</i>
---	--

השתמשו בקובץ השלד המצורף לתרגיל hw4q3.c. קובץ השלד כמובן מתאים למקרה שבו בכל השאלות התשובה הנכונה היא 'a'. מה שעליכם לעשות הוא לשנות את האות 'a' לאות המתאימה לתשובה שלכם בכל אחד מששת הסעיפים. שימו לב, שלכל קטע קוד יש לבחור גם את האות המתאימה לסיבוכיות זמן (time complexity) וגם את האות המתאימה לסיבוכיות מקום (space complexity). אנא עיינו בקבצי הקלט המסופקים לתרגיל.

### בהצלחה