

## מבוא למדעי המחשב תרגיל בית 5

נושא: מחרוזות, מצביעים, הקצאות

מתרגלים אחראים: אמיר ישורון ותמר ריזמוביץ

### הערות כלליות:

המטלה כוללת 3 קבצים:

– mainTrain.c קובץ ה- main שלכם, הוא מריץ בדיקות על הפונקציות שתממשו.

– Util.h קובץ הגדרות עבור הפונקציות.

– Util.c קובץ מימושים עבור הפונקציות.

במטלה זו תתבקשו לממש 4 פונקציות. המימושים צריכים להיות בתוך ההגדרות שמוכנות

לכם בקובץ Util.c

לאחר שתממשו, תוכלו לקמפל ולהריץ את התכנית עם ה- main -המוכן או לשנות את ה- main - כראות עיניכם ולבצע עוד בדיקות משלכם.

בסיום המטלה הנכם נדרשים להגיש **קובץ אחד בלבד** - Util.c ובו המימושים של השאלות.

לא לשכוח לרשום הערות לאורך הקוד (כל 2-3 שורות).

כמו כן הנכם מתבקשים לרשום הערה ארוכה בתחילת התכנית הכוללת שם, ת.ז, תאריך ומה התכנית עושה.

## מחרוזות ללא פויינטרים:

### שאלה מספר 1:

א. כתבו פונקציה המקבלת מחרוזת של מילים וממיינת את המילים לפי סדר לקסיקוגרפי (הכוונה היא שהפונקציה ממיינת את המטריצה matStr והיא איננה מקצה מטריצה חדשה).  
אין צורך להדפיס את הפלט):

```
void SortByAB(char matStr[][COLS], int rows, int cols);
```

לדוגמא עבור מחרוזת המילים הבאה:

Input:

yehuda

galit

haim

aviv

inbal

Output:

aviv

galit

haim

inbal

yehuda

ב. כתבו פונקציה המקבלת מחרוזת של מילים וממיינת את המילים לפי אורך כל מילה מהקטן לגדול (אם האורכים זהים, אין חשיבות לסדר הופעת המחרוזות בתוצאה. גם כאן אין צורך להדפיס את התוצאה של המיון):

```
void SortByLenght(char matStr[][COLS], int rows, int cols);
```

לדוגמא עבור מחרוזת המילים הבאה:

Input:

yehuda

galit

haim

aviv

inbal

Output:

aviv

haim

galit

inbal

yehuda

## פויינטרים:

### שאלה מספר 2:

כתבו פונקציה המקבלת מצביע למערך של int-ים, משתנה int המכיל את גודלו של המערך, מצביע ל-double ומצביע ל-int.

חתימת הפונקציה נראית כך:

```
void AnalyseArray(int* arr, int size, double* avg, int* startIdx);
```

המשתנה size מכיל את גודלו של המערך arr.

הפונקציה בודקת מה היא תת הסדרה היורדת הארוכה ביותר במערך arr. היא מעדכנת בערך המוצבע על ידי avg את ממוצע תת הסדרה ובערך המוצבע על ידי startIdx את האינדקס של תחילת הסדרה.

### לרשותכם הדוגמא הבאה:

בהינתן המערך arr הבא:

1	5	-9	-78	9	5	4	3	8	100
---	---	----	-----	---	---	---	---	---	-----

התת סדרה היורדת הארוכה ביותר היא:

9	5	4	3
---	---	---	---

הערך הממוצע של הסדרה הוא 5.25, והערך שייכנס ל-startIdx הוא 4 (האינדקס של 9 במערך המקורי הוא 4).

הערות:

- במקרה שישנן מספר סדרות יורדות בעלות אותו האורך, יש להחזיר את התוצאות עבור הסדרה הראשונה שמופיעה.
- איבר בודד הוא סדרה יורדת בפני עצמו באורך אחד.
- כאשר יש שני איברים זהים ברצף, הרצף איננו נחשב רצף יורד.

## פויינטרים:

### שאלה מספר 3:

כתבו פונקציה המקבלת מחרוזת str, תו (char) separator, מערך של char\* בשם words ומערך של int-ים בשם lengths. ערך החזרה של הפונקציה הוא int.

חתימת הפונקציה נראית כך:

```
int Separate(char* str, char separator, char** words, int* lengths);
```

הפונקציה תשבץ במערך words את הכתובות של תחילת המילים במחרוזת str, כאשר המילים מופרדות על ידי התו separator. כמו כן, המערך lengths יכיל את מספר התווים בכל מילה לפי סדר הופעתן. ערך החזרה של הפונקציה הוא מספר המילים הכולל במחרוזת str.

שימו לב כי אין צורך להקצות את המערכים words ו-lengths. יש להניח כי הם מכילים די מקום לאחסון המידע כולו.

### לרשותכם הדוגמא הבאה:

בהינתן המחרוזת str (השורה הראשונה היא אותיות המחרוזת, והשורה השניה היא הכתובת של כל תו):

C	o	l	m	a	n		l	n	t	r	o	-	t	o	-	C	S	\0
70	70	70	70	70	70	70	70	70	70	71	71	71	71	71	71	71	71	71
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8

וה-separator "-" (התו -), המערך words יכיל את הערכים הבאים:

700	713	716
-----	-----	-----

700 היא הכתובת בזיכרון של התו C, שהיא המילה הראשונה עד ה-"-". הראשון.

המילה הבאה היא to, המתחילה בכתובת 713, והמילה האחרונה היא CS שמתחילה בכתובת 716.

המערך lengths יכיל את הערכים הבאים:

12	2	2
----	---	---

שימו לב כי התו רווח איננו משמש כמפריד בין מילים, אלא אך ורק התו המתקבל ב-separator.

ערך החזרה של הפונקציה הוא 3, שכן המחרוזת הכילה 3 מילים.

### הערות:

- יתכן מצב בו התו separator יופיע פעמיים ברצף.
- ניתן להניח כי הקלט יכיל תווים נוספים שאינם separator, וכן ש-separator לא יופיע כתו האחרון במחרוזת הקלט.

### המלצה:

נסו שלא להשתמש בסוגריים מרובעים בעת פתירת התרגיל, זה יותר פשוט כך.

## הקצאות:

### שאלה מספר 4:

כתבו פונקציה המקבלת מחרוזת המכילה מילים המופרדות ברווח, ומחזירה מחרוזת חדשה, שבה סדר המילים הוא סדר הפוך.

חתימת הפונקציה:

```
char* ReverseWords(char* words);
```

לדוגמא:

```
input: "There was a cat"
```

```
output: "cat a was There"
```

```
input: "cat"
```

```
output: "cat"
```