

מבוא למדעי המחשב מ'ח' (234114/7), סמסטר חורף 2015/6

תרגיל בית 4

מועד אחרון להגשה: יום רביעי 6.1 בשעה 23:00

המתרגל האחראי על תרגיל זה: **יותם אשל**

משרד: טאוב 315

E-mail: yotam.happy@gmail.com

שעת קבלה רגילה: יום ג' 15:30-16:30

שעות קבלה מיוחדות לשאלות על התרגיל: ימי ראשון 27.12 ו 3.1 בשעה 15:00
אם שעות אלו אינן נוחות לכם ניתן לתאם פגישה בשעות אחרות.

הנחיות כלליות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרון.
- שאלות ותשובות נפוצות בנוגע לתרגיל יתפרסמו באתר כל כמה זמן תחת סעיף F.A.Q.
- **חובה להיכנס ולהתעדכן מדי פעם! כל דגש שמפורסם שם הוא מחייב!**
- 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 להודיע באי-מייל למתרגל האחראי לתרגיל זה ולשלוח לו עותק של האישור המראה שהייתם במילואים (טופס 3010).
- אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל.
- ערעורים על הבדיקה הידנית ניתן להגיש במייל למתרגל האחראי על התרגיל **עד שבוע לאחר קבלת הציון**. לא ניתן לערער על כמות הנקודות שהורדה בגין טעות מסוימת!
- לא ניתן לערער על תוצאות הבדיקה האוטומטית אלא אם הקוד שהגשיתם עובר את כל הבדיקות באתר הבדיקות האוטומטי ולמרות זאת נכשל באותן הבדיקות בדיוק (שהן ארבע הבדיקות הראשונות) בבודק האוטומטי.
- בתרגיל זה מותר להשתמש בפונקציות מהספרייה `stdio.h`, `stdbool.h`, **בלבד**, שנלמדו בהרצאות ובתרגולים (מלבד במקום המצוין בפירוש אחרת). כמו כן, החומר המותר לתרגיל הוא כל החומר הנלמד עד הרצאה 9 (כולל) ועד תרגול 10 (כולל).

פונקציות בקוד:

בדומה לתרגיל 3, יש להקפיד על חלוקת הקוד לפונקציות, הערות, ועל קריאות הקוד. בתרגיל זה אורך כל פונקציה צריך להיות קטן מ 22 שורות (כמו במבחן, ולא 16 כמו בתרגיל 3).

שאלה 1 – חישוב במספרים גדולים

לשם כתיבת תוכנה לחישוב מדעי רוצים לממש תוכנית המסוגלת לבצע פעולות חשבון במספרים גדולים – עד 30 ספרות. מאחר ואין טיפוס נתונים ב C המסוגל לטפל במספרים באורך כזה הוחלט להשתמש במערך של ספרות לייצוג המספרים. על התוכנית לאפשר ביצוע שתי פעולות מתקדמות: סכימה של n מספרים וכפל של n מספרים.

- עלינו לייצג מספר בן 30 ספרות. נשים לב כי כל המספרים במערכת חיוביים בלבד.
- דוגמאות:
המספר 345678901234567 מיוצג על ידי מערך באורך 30 תאים באופן הבא:
(7,6,5,4,3,2,1,0,9,8,7,6,5,4,3,0,0, ..., 0)
המספר 987654321 מיוצג ע"י מערך באורך 30 תאים באופן הבא:
(1,2,3,4,5,6,7,8,9,0,0, ..., 0)
- שימו לב כי הספרות מיוצגות במערך מהפחות משמעותית ליותר משמעותית. כלומר ספרת האחדות ראשונה, העשרות שניה, המאות שלישית וכן הלאה.
- במקרה של *overflow* נדווח כי ארעה גלישה (והמספר שנחזיר יכול להיות זבל – אין חשיבות למה יהיה בו)

(א)

עליכם לממש פונקציה עם החתימה הבאה:

```
void calcBig(char op, unsigned char operands[][MAX_DIGITS], int numOperands, unsigned char result[MAX_DIGITS], bool* overflow);
```

- *Op* – האופרטור (פעולה) שעלינו לחשב. יכול להיות '+' אם רוצים שנחשב חיבור. '*' אם רוצים שנחשב כפל
- *Operands* – מערך של מספרים 'גדולים' עליהם מתבצעת הפעולה. כל מספר הוא מערך בעצמו, לכן זהו מערך דו מימדי.
- *numOperands* – מספר האופרנדים (המספרים עליהם מתבצעת הפעולה)
- *result* – מערך לתוכו יש לכתוב את תוצאת הפעולה
- *Overflow* – כתובת משתנה בוליאני לתוכו יש לכתוב *true* אם היתה חריגה (*overflow*) ו *false* אם לאו.
- במקרה של חיבור על הפונקציה לרוץ בסיבוכיות זמן $O(nk)$ כאשר n מספר האופרנדים. ו k אורך מקסימאלי לכל מספר (זהו אומנם קבוע 30 אבל נתייחס אליו כמשתנה). במקרה של כפל על הסיבוכיות להיות $O(nk^2)$

(ב)

עליכם לכתוב תוכנית המשתמשת בפונקציה שכתבתם.

- עליכם לקלוט מהמשתמש את האופרטור (הפעולה) שרוצים לחשב: '+' או '*'. לאחר מכן לקלוט מהמשתמש על כמה אופרנדים ברצונו לבצע את הפעולה (מובטח כי קטן מ 20) לאחר מכן על התוכנית לקלוט את האופרנדים (המספרים עליהם מתבצעת הפעולה). האופרנדים בני לכל היותר 30 ספרות, כל אחד בשורה חדשה. ניתן להניח שהקלט תקין ואין צורך לבצע בדיקות תקינות. לבסוף יש לבצע את החישוב ולהדפיס את התוצאה.
- **שימו לב** כי כאשר קולטים את המספרים חובה לקלוט באמצעות `scanf("%s", str)`. (אומנם מפתה לקרוא את הספרות אחת אחת ע"י `%c` אולם זוהי פעולה עדינה שמועדת לטעויות)

דוגמאות הרצה:

```
Select operator:
*
How many operands:
4
Enter operands (seperated by new line):
2
3
4
5
result:
120
```

```
Select operator:
+
How many operands:
2
Enter operands (seperated by new line):
99
11
result:
110
```

[illegible]

```
Select operator:
*
How many operands:
3
Enter operands (seperated by new line):
1234567890
1234567890
1234567890
result:
1881676371789154860897069000
```

שאלה 2 - Sentiment Analysis

לאחר שנים שהוליווד מפיקה סרטי קומיקס עתירי תקציב בקצב שעולה על יכולת התסריטאים לאתר עוד ועוד חוברות קומיקס נידחות הוחלט לבדוק מה הקהל באמת חושב על הסרטים האלו. לשם כך ביקשו כמה מנהלים בכירים לכתוב מערכת שמזהה האם ביקורות על סרטים הן חיובית או שלילית. עליכם הוטלה בניית המערכת.

הגדרת המערכת:

1. המערכת קולטת משפט או טקסט קצר.
2. למערכת 3 מילונים פנימיים. הראשון מילון של מילים חיוביות, השני של מילים שליליות והשלישי של מילות היפוך. להלן המילון:

Positive	delight	amaz	Excit	Wonderful	Great	Lov	Stun	good
Negative	repuls	Bad	Horribl	Disappoint	Hat	Bor	Annoy	
Invert	Not	Isn't	Wasn't	Opposite				

3. המערכת מחפשת מופעים של מילים מתוך המילונים השונים ומחשבת את ציון ה sentiment של הקלט על בסיס זה. אולם לפני שנסביר את החישוב יש צורך להסביר כיצד מתבצעת התאמה של מילה למילון:
 - **התאמה של מילה למילון הינה case insensitive**. לדוגמא:
Not==not==NOT
 - **מספיק להתאים את המילה במילון לתחילית של מילה מהקלט**. לדוגמא
Amaz מהמילון יכול להתאים לכל אחת מהמילים הבאות בקלט:
Amaz,amazing,amazement,AmAziNG!!
4. **הציון של ביקורת** יוגדר כמספר המילים החיוביות פחות מספר המילים השליליות, כאשר אם מופיעה מילת היפוך אחת לפחות מכפילים ב -1.

אזי מה שעליכם לעשות:

א. כיתבו פונקציה המקבלת מילה ומילון ובודקת האם המילה קיימת במילון. להלן החתימה של הפונקציה:

```
bool matchList(char *str, char **wordList, int wordListLen);
```

- אין לשנות את החתימה. הפונ' מחזירה האם נמצאה התאמה מהמילון או לא.
- הפרמטרים שהפונקציה מקבלת:
 - `str` – המילה שיש לבדוק האם מתאימה למילה במילון
 - `wordlist, wordListLen` – מערך של מחרוזות ואורכו. זהו המילון. ניתן להניח כי המילון ממוין לפי סדר מילוני (לקסיקוגרפי)
- **ההתאמה תעשה לפי הכללים שהוגדרו למעלה: ההתאמה היא case-insensitive ומספיק להתאים מילה מהמילון כתחילית בלבד.**
- **סיבוכיות זמן ריצה:** אם נסמן ב n את האורך של `str`, וב k את מספר המילים במילון, **על הפונקציה לרוץ בסיבוכיות זמן של $O(n \log(k))$.** כמו כן סיבוכיות המקום הינה $O(1)$, כלומר הפונ' משתמשת בזיכרון קבוע ללא קשר ל n ו k . תכננו היטב כיצד לבצע זאת.
- **טיפ:** כתבו קודם את הפונקציה כך שמתאימה מילים זהות בדיוק בסיבוכיות הרצויה, אחר כך הוסיפו case-insensitive ולבסוף בצעו שינוי קטן כך שתתאפשר התאמת תחיליות בלבד.

ב. כתבו תכנית הקולטת ביקורת, ובעזרת הפונ' שמימשתם בסעיף א' פולטת את הציון שלה. הוראות:

- התוכנית תקלוט קלט עד אשר נקלט EOF. **אין הגבלה על אורך הקלט.** אולם נתון כי **אורך מילה לא יעלה על 30 תווים. מילה מוגדרת:** כרצף של אותיות אשר משני צדדיו רווח, טאב או שורה חדשה.
- אם נסמן n אורך מילה מקסימלי, k מספר המילים בשלושת המילונים ו m מספר המילים בקלט, אזי סיבוכיות זמן הריצה הכולל של התוכנית תהיה:
 $O(m \cdot n \cdot \log(k))$
- סיבוכיות המקום תהא $O(n+k)$ בלבד.
- אם התוצאת החישוב היא חיובית אזי יודפס:
Sentiment score: <value>; Positive review
- אם שלילית: **Sentiment score: <value>; Negative review**
- אם שווה אפס: **Sentiment score: 0; Couldn't tell**
- נא להקפיד על שורה חדשה ($\backslash n$) בסוף כל שורת פלט. גם האחרונה.

דוגמאות הרצה:

```
Enter a sentence to evaluate:  
i was LOVING this movie, it was soooo good!  
^Z  
Sentiment score: 2; Positive review
```

```
Enter a sentence to evaluate:  
plot was disappointing but camera work was stunning. actors were good too  
^Z  
Sentiment score: 1; Positive review
```

```
Enter a sentence to evaluate:  
This wasn't a good movie and WASN'T great at all!  
^Z  
Sentiment score: -2; Negative review
```

```
Enter a sentence to evaluate:  
It had delightful performances but many boring moments...  
^Z  
Sentiment score: 0; Couldn't tell
```

דגשים נוספים:

- בכל פעם שאתם מתבקשים להדפיס שורה כלשהי בתכנית, ההדפסה מסתיימת בירידת שורה בודדת ללא רווח בסוף (אפילו אם מדובר בשורה האחרונה בתכנית!). אין להדפיס רווחים שלא התבקשתם להדפיס (בפרט בתחילת שורה או בסופה) ויש לשים לב לאותיות גדולות וקטנות לפי הדוגמאות.
- עליכם לוודא את נכונות הקוד שלכם בשאלות 1,2 באמצעות אתר הבדיקה בכתובת <http://cs12.cs.technion.ac.il/~cs234114>. גם הפעם מסופקים לכם באתר הקורס קבצי הקלט והפלט המצופה לטסטים שבאתר הבדיקה.

הוראות הגשה:

- ההגשה הינה אלקטרונית בלבד 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 וכתובת האי-מייל שלך.
 - קובץ פתרון `hw4q1.c` עבור שאלה 1.
 - קובץ פתרון `hw4q2.c` עבור שאלה 2.
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה לא תתקבל ע"י המערכת!
- חובה לשמור את אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס. שימו לב כי יש לשמור את כל הפרטים המופיעים באישור (שם הקובץ, תאריך ההגשה וכו') ולא רק את המספר!

בהצלחה !