Bar-Ilan University
אוניברסיטת בר-אילן

מיקרו-מעבדים ושפת אסמבלר 83-255

תשפ"א/2021 - תרגיל מס' 4

אחראי תרגיל - אלדור

חלק א – תרגיל יבש

נתונה השגרה הבאה:

```
0026              VEC_MUL PROC NEAR
0026  59                POP CX
0027  5E                POP SI
0028  5F                POP DI

                        ; Do the work
0029  33 C0             XOR AX, AX
002B  8A 04             MOV AL, [SI]
002D  F6 2D             IMUL BYTE PTR [DI]     ; Multiplication output in AX

                        ; Base Case
002F  83 F9 01          CMP CX,1
0032  75 01             JNZ REC
0034  C3                RET

                        ; Recursion Case
0035                    REC:
0035  50                     PUSH AX     ; save for later

0036  46                     INC SI
0037  47                     INC DI
0038  49                     DEC CX
0039  57                     PUSH DI
003A  56                     PUSH SI
003B  51                     PUSH CX
003C  E8 FFE7                CALL VEC_MUL

003F  5B                     POP BX      ; restore saved AX
0040  03 C3                  ADD AX, BX
0042  C3                     RET
0043              VEC_MUL ENDP
```

להלן קוד לדוגמא שקורא לשגרה:

```
.data
VEC1        db 1,-1,1,-1,1
VEC2        db 1,5,1,4,1
LEN         dw 5

.code
.startup
      mov ax, offset VEC1
      push ax
      mov ax, offset VEC2
      push ax
      mov ax, LEN
      push ax
      CALL VEC_MUL
      …
```

א. הסבירו מהי מטרת השגרה VEC_MUL.

ב. מהי שיטת העברת הפרמטרים אל השגרה ? כמה פרמטרים מועברים אליה ומה תפקדי כל אחד מהפרמטרים ?

ג. מהי שיטת העברת הפרמטרים חזרה מהשגרה ? כמה פרמטרים מועברים ומה תפקיד כל אחד מהפרמטרים ?

ד. סטודנט נמרץ ניסה להפעיל את הקוד הנ"ל וגילה כי הוא לא עובד. באמצעות תוכנת ניפוי שגיאות (דיבאגר) גילה הסטודנט כי קיימת בעיה בהעברת הפרמטרים אל השגרה: מיד עם הפעלת השגרה (לפי ה"קוד לדוגמא" לעיל) האוגר CX מאותחל לערך שגוי ולא לגודל 5 כצפוי.

a. הסבירו מדוע CX אינו מאותחל לערך הנכון.

b. כתבו קטע קוד קצר אשר פותר את הבעיה של העברת הפרמטרים אל השגרה.
ציינו במפורש אילו שורות בקוד לעיל יש להחליף עם קטע הקוד שכתבתם.

ה. מה המשמעות של ההנחיה BYTE PTR בשורה 002D המסומנת ? מדוע נדרש להוסיף הנחיה זו למהדר בשורה זו ?

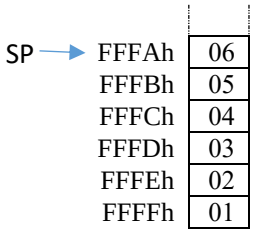
חלק ב – תרגיל רטוב

1. הדפסת המחסנית למסך:

כתבו שגרה המדפיסה למסך את תוכן המחסנית.
השגרה מקבלת כקלט את מספר הבתים שיש להדפיס (החל מראש המחסנית) ומציגה על המסך את תוכן המחסנית **שלפני הקריאה לשגרה**, בפורמט הקסדצימלי, באופן טורי, בצורה כזו שתשקף את תוכן המחסנית, כאשר המספר האחרון שיוצא מהמחסנית, יודפס בתחתית.

הקלט לשגרה יתקבל באוגר ax. כלומר השגרה מניחה שהפרמטר הוכן שם מראש.

לדוג', עבור קריאה לשגרה עם הקלט 4, כאשר תוכן המחסנית **לפני** הקריאה לשגרה היא כפי שמצורף באיור, יודפס למסך:



06
05
04
03

<u>הערות ודגשים:</u>

- עליכם להכניס למחסנית באופן ידני, לפני הרצת השגרה ,לפחות 6 ערכים. ז"א שax יקבל ערכים בין 1 ל-6.
- ניתן להניח שלא יינתן ערך גדול מ20. כלומר מקסימום 20 ערכים יודפסו.
- מומלץ להיעזר בקוד שמימשתם בתרגיל הבית הקודם.
- לקובץ הקוד יש לתת את השם: **PrintStk.asm**.

2. LCM – Longest Common Subsequence:

בתרגיל זה עליכם לממש את האלגוריתם הרקורסיבי לפתרון בעיית ה-LCM המצורף בעמוד הבא.

רקע:

קלט הבעיה: 2 מחרוזות M ו-N בעלות אורכים m ו-n כלשהם בהתאמה.

פלט הבעיה: תת המחרוזת המשותפת הארוכה ביותר ואורכה.

לקריאה נוספת: https://en.wikipedia.org/wiki/Longest_common_subsequence_problem.

https://www.geeksforgeeks.org/longest-common-substring-dp-29/.

הנחיות למימוש הקוד:

- בזיכרון הדאטא ישמרו 2 ה"מחרוזות", לדוגמא:

```
M db 'Message 1 goes here!'
N db 'Message 2 will be here :)'
```

- **בונוס -** את מחרוזת הפלט עליכם לשמור בזיכרון הדאטא, לדוגמא עבור ה-M וה-N הנ"ל עליכם לשמור במיקום בזיכרון שתגדירו מראש את המחרוזת:

```
'Message  e here'
```

- הפרמטרים המועברים לשגרה הרקורסיבית הם:
  - אורך מחרוזת א'.
  - אורך מחרוזת ב'.
  - אורך תת המחרוזת המשותפת הארוכה ביותר (ה-LCM).
- הפרמטר שמוחזר מהשגרה הינו אורך תת המחרוזת הארוכה ביותר.

הנחיות לחלק היבש: (הגשה כ-PDF)

a. הכינו תרשים זרימה עקרוני של התוכנית כפי שראיתם בכיתה.
b. רשמו מה תפקידו של כל אוגר.
c. מה אורך המחרוזות הארוך ביותר שהקוד מבטיח לתמוך בו בהנחה וברשותנו סגמנט אחד לזיכרון המחסנית (64k כתובות) ? הסבירו.
d. מה סיבוכיות הזמן והזיכרון (דאטא + מחסנית) של האלגוריתם הרקורסיבי ? האם יש פתרון אחר בעל סיבוכיות זמן וזיכרון (דאטא + מחסנית) קטן יותר ?

הערות ודגשים נוספים:

- ודאו שאתם מבינים איך עובד האלגוריתם הרקורסיבי לפני תכנון הקוד וכמובן לפני כתיבת הקוד.
- לא מומלץ כלל לנסות "לתרגם" קוד משפה עילית לאסמבלר.
- לקובץ הקוד יש לתת את השם: **LCM.asm**.

האלגוריתם למימוש:

```
int lcs(int i, int j, int count) {
    if (i == 0 || j == 0)
        return count;

    if (X[i - 1] == Y[j - 1])
        count = lcs(i - 1, j - 1, count + 1);
    else
        count = count + max(lcs(i,j-1,0), lcs(i-1,j,0));
    return count;
}
```