תרגילי אתגר (בסגנון תחרות קודגורו אקסטרים)

ברק גונן, מבוסס על תרגילים מאת יוסי זהבי

רקע: למדנו לבצע פעולות העתקה באמצעות רגיסטרים או באמצעות סגמנט הנתונים בזיכרון, אשר ds מצביע עליו. כעת נתנסה בפעולה לא שגרתית- שינוי קוד התוכנית תוך כדי ריצה. כדי לבצע שינוי בקוד תוך כדי ריצה צריך לפנות למקום הנכון בזיכרון- לסגמנט הקוד.

נראה דוגמה לפניה לסגמנט הקוד:

```
mov   [byte ptr cs:0000h], 0AAh
```

פעולת העתקה זו מעתיקה את הערך 0AAh לתוך הבית שנמצא בכתובת הבאה: סגמנט הקוד, אופסט 0000h.

אם ברצוננו להעתיק מילה (ולא בית יחיד), נשנה את הפקודה כך שתכלול את ההוראה word ולא byte. לדוגמה:

```
mov   [word ptr cs:0000h], 0AABBh
```

עד כאן ההסבר... מכאן נסו להסתדר לבד.

רמת הקושי בשאלות עולה.

1. כתבו תכנית שתאחסן בתא הזיכרון הראשון של מקטע הקוד את הערך 0CCh.
2. כתבו תכנית שתאחסן בסוף התוכנית (שורה לפני הפקודה mov ax, 4C00) את הערך 0CCh.
3. כתבו תוכנית שמעתיקה לתוך al את הערך 1. נראה קל מדיי? עשו זאת כמובן בלי להשתמש ברגיסטר al או ax בתכנית.
4. לפניכם קטע קוד המאפס את ax ולאחר מכן קורא 4 פעמים ל add al,2. העתיקו את קטע הקוד הזה לתכנית שלכם בלי להכניס בו שינויים כלשהם (מותר להוסיף שורות קוד לפניו...). גירמו לכך שעם ההגעה לתווית "here" ערכו של al יהיה 7.

```
mov ax,0
add al,2
add al,2
add al,2
add al,2
here:
```

5. שוב נתון אותו קטע קוד המאפס את ax ולאחר מכן מעדכן את ערכו של al. גם הפעם העתיקו אותו לתכנית שלכם בלי לבצע שינויים כלשהם. כעת גירמו לכך שעם ההגעה לתווית "here"ערכו של al יהיה 4, ואילו ערכו של ah יהיה 2.

רמז: בשלב מסוים תוכלו להשתמש בפקודה nop, קיצור של No Operation, פקודה שאינה מבצעת כלום.

```
mov ax,0
add al,2
add al,2
add al,2
add al,2
here:
```

6. צרו תוכנית ובה שורת הקוד:

```
here:  mov   ax, 0
```

a. באמצעות הוספה של פקודות mov בלבד, גרמו לכך שהתוכנית לעולם לא תגיע לתווית here (עליכם להכיר את פקודת jmp, עליה ניתן לקרוא בספר הלימוד)

b. שוב באמצעות פקודות mov בלבד, גירמו לכך שהתוכנית תריץ לפני התווית here לולאה אינסופית וספרו את כמות הפעמים שהלולאה רצה באמצעות הרגיסטר cl.

7. תרגיל הצפנה (קרדיט: **אלדד קפיטולניק**): ברכיבים מסויימים יש צורך שהמידע יישמר בזיכרון בצורה מוצפנת, כדי שלא ניתן יהיה לקרוא אותו משם ולפענח אותו. לדוגמה, קודי גישה שנמצאים על כרטיסי זיהוי חכמים, כרטיסי סים של טלפונים סלולריים ועוד. מטרת התרגיל הבא היא ליצור תוכנית ששומרת מידע בזיכרון בצורה מוצפנת. לצורך יצירת ההצפנה- קיראו על הפקודה XOR בספר הלימוד.

כמו בתרגילים הקודמים, את הקוד לביצוע התוכנית יש ליצור תוך כדי ריצה. ניתן לשתול פקודות nop בתכנית הראשית, כהכנה לקוד שייכתב במהלך הריצה, אך מומלץ לשתול פקודת jmp.

- בשלב הראשון יש לשתול באופסט 100h עד אופסט 103h (בסגמנט הנתונים) קוד סודי בן 4 ספרות (בייצוג ASCII של הספרות- לדוגמה הקוד 1234 יהיה 31h 32h 33h 34h)
- בשלב השני - יש להוסיף לקוד קטע שמעתיק את הקוד הסודי לאופסט אחר בזיכרון (כתובות 104h-107h) אך הנתונים מועתקים בצורה מוצפנת .
- בשלב שלישי - יש להוסיף קטע קוד שמפענח את הקוד הסודי המוצפן ומעתיק את התוצאה לכתובות 108h-111h.