

מיקרו-מעבדים ושפת אסמבלר 83-255

תשפ"א/2021 - תרגיל מס' 2

חלק 1 – תרגיל יבש

- (1) עבור כל אחת מהפקודות הנ"ל, כתבו האם הפקודה חוקית או לא.
 אם כן – מהו ה *Opcodes* המתאים לפקודה.
 אם לא – הסבירו מדוע הפקודה אינה חוקית.

<i>Command</i>	<i>Explanation\Opcode</i>
<code>mov cx,[bx + 8]</code>	
<code>mov cx,[cx + si]</code>	
<code>mov dx,[bx + si]</code>	
<code>mov ds,[bx + si]</code>	
<code>mov dx,[bx - si]</code>	
<code>mov dx,[bx - 2 * 200]</code>	
<code>mov dx,[bx + 4 * si]</code>	

- (2) עבור ה *Opcodes* המתאימים, תרגמו את הפקודות:

<i>Opcode</i>	<i>Command</i>
FEC7	
B80200	
83C10F	
B65E	
2C0A	
BB5005	

(3)

א. תנו 3 דוגמאות לכתובות השקולות לכתובת **1357h:2468h**.

ב. הטבלה שלפניכם מייצגת תכנית אסמבלי.

i. השלימו את ההערות

ii. הסבירו בקצרה מה התכנית עושה.

(שימו לב להסבר אודות הפקודה LOOP L2)

Address	Machine language	Instruction in assembly language*	Comments
EB8:0100	B800B8	MOV AX, 0B800h	
EB8:0103	8ED8	MOV DS,AX	
EB8:0105	B025	MOV AH, 30h	
EB8:0107	BB0104	MOV BX, 07D0h	
EB8:010A	B90500	L1: MOV CX, 10d	
EB8:010D	8807	L2: MOV [BX], AH	
EB8:010F	83C302	ADD AH, 01h	This instruction adds 1 to the value of AH. AH <- AH+1
EB8:0112	E2F9	LOOP L2	This instruction causes the CPU to repeat the code that starts at address 010D (label L2) and ends at this line. The number of repetitions is determined by the value of CX (set at label L1).
EB8:0114 EB8:0116	B4 4C CD 21	MOV AH, 04Ch INT 21h	Returns control of the program to the operating system

*all the numbers are represented in the hexadecimal method.

מה התכנית מבצעת?

חלק 2 – תרגיל רטוב

שאלה מספר 4

בתרגיל זה אנחנו נבנה תכנה המדפיסה ציור השמור בזיכרון.

לצורך תרגיל זה מצורף קובץ אסמבלי המכיל בתוכו מידע ששמור בזיכרון וכן קטע קוד שתוכלו להיעזר בו (לא חובה להשתמש בו).

א. בסעיף זה עליכם להדפיס למסך בעזרת שימוש בפקודות MOV את הציור ששמור במשתנה AsciiArt1.

- כל איבר במערך הוא ייצוג תו בפורמט ascii אותו יש להציג על המסך (את הרקע והצבע קבעו כרצונכם – נספח מופיע בהמשך).
- המידע שמור בסדר הכתובות של המסך, כלומר הערך הראשון שייך למיקום השמאלי העליון, השני שייך למיקום השני העליון וכן הלאה.

ב. בסעיף זה עליכם להדפיס למסך את התמונה במשתנה AsciiArt2.

כחלק מאופטימיזציית זיכרון, המערך הזה אינו מכיל תווי ASCII כמו בסעיף הקודם. במקום, המערך מכיל "קוד" של תווים בפורמט אחר – BASKI. קוד ה-BASKI מכיל 16 תווים בלבד, לפי הפירוט שבמערך הבא:

```
Dictionary db ' ', '-', '/', '\', '|', '(', ')', '>', '<', '"', '.', ',', '-.', '!', '+', ']', '!',
```

לדוגמא: אבר בתמונה בעל הקוד 0, משמעותו – יש להדפיס רווח. אבר בתמונה עם הקוד 1, משמעותו – יש להדפיס מקף. וכו'.

כתבו קוד שמדפיס למסך את התמונה AsciiArt2 כך שהוא נראה בצורה תקינה.

הדרכה: השתמשו במשתנה Dictionary על-מנת לבצע את התרגום מקוד BASKI לקוד ASCII

ג. בסעיף זה עליכם לכתוב קובץ ASM שמכיל:

- (1) משתנה בשם BaskiArt1 המכיל את התמונה AsciiArt1 כאשר היא מקודדת לפי קוד BASKI
- (2) קוד שמציג את התמונה שבמשתנה BaskiArt1 על המסך (ניתן להשתמש בקוד מסעיף ב')

הדרכה: היעזרו בשפה עלית כגון c, python וכו'.

הערות ודגשים:

1. שמירת משתנים בצורה הבאה:

```
50 ,DB 10 TEMP
```

שקולה לצורה:

```
TEMP DB 10
      DB 50
```

בקובץ המצורף השתמשנו בשיטה השניה כי הקומפילר איתנו אנחנו עובדים (ה-masm) מגביל את אורך התווים בכל שורה (עד ה-enter) ומייצר לעיתים שגיאה כשיש שורות ארוכות.

2. על מנת להקל עליכם הוספנו בקובץ המצורף קוד של לולאה שמבצעת 2000 חזרות – כמספר הסימנים על המסך. הפקודה loop בה השתמשנו עובדת עם האוגר CX ולכן אם אתם משתמשים בלולאה הזו אסור לקוד שלכם לשנות את אוגר CX.

3. קישור לטבלת ASCII - <http://www.asciitable.com>

4. נספח עזר לכתיבה למסך:

Memory location

Text displayed on the screen is stored in the framebuffer. The location of the frame register depends on the definition of the text color: black and white or color:

Emulation	Frame buffer physical address	Frame buffer real-mode address
color	B8000h	B800h:0000
monochrome	B0000h	B000h:0000

Every character in the framebuffer takes up 2 bytes: one byte for the Character at location N, and one byte for the Attribute at location N+1.

bit in attribute byte		Usage	
Bit 7		Blink	
Bit 6- bit 4		background color (0-7)	
Bit 3- bit 0		foreground color (0-15)	
color value	color	color value	color
0	black	8	dark gray
1	blue	9	light blue
2	green	10	light green
3	cyan	11	light cyan
4	red	12	light red
5	magenta	13	light magenta
6	brown	14	yellow
7	light gray	15	white