

מיקרו-מעבדים ושפת אסמבלר 83-255

תשפ"א/2021 - תרגיל מס' 3

אחראי תרגיל - אלדור

חלק א – תרגיל יבש

1. נתון חלק התוכנית הבא:

- 1) `mov bx, 0FFh`
- 2) `xor ax, ax`
- 3) `dec ax`
- 4) `sub ax, bx`
- 5) `test ax, bx`
- 6) `mul bx`
- 7) `not bx`

מתוך אוגר הדגלים נתון שמצב הביטים לפני ביצוע פקודה 1 הינו:

OF	SF	ZF	AF	PF	CF
0	1	0	1	0	1

כתבו את מצב ביטים אלו לאחר שורות פקודה 3, 5 ו-7.

הערות ודגשים:

- עבור פקודה שבה אוגר הדגלים אינו מוגדר היטב יש לכתוב '?'.
עבור ביטים שלא השתנו ניתן לציין זאת מבלי לכתוב את הערך המספרי מפורשות.
- המלצה - עקבו אחר הערכים של האוגרים וציינו אותם בצורה מפורשת כ-4 ספרות הקסדצימליות.

2. נתונה תבנית הקוד הנ"ל:

- 1) `mov dx, **`
- 2) `mov cx, **`
- 3) `L1: ROL **, 1`
- 4) `jnc L2`
- 5) `inc **`
- 6) `L2: dec **`
- 7) `jnz L1`

עליכם להשלים את הפקודות במקומות המסומנים, כך שהתוכנית תבצע את הנ"ל:
תספור את מספר הסיביות שערך 1 באוגר BH, התוצאה לכך תהיה באוגר DX. (שימו לב שיש שימוש ב-CX כאוגר עזר).

חלק ב – תרגיל רטוב

1. הדפסת אוגר למסך:

- 1) עליכם להדפיס למסך את התוכן של האוגר AX בייצוג הקסדצימלי (בסיס 16).
לדוגמא עבור $AX = 1F2Ch$ יודפס למסך $AX = 1F2Ch$.

הערות ודגשים:

- בסעיף זה אין להשתמש בפקודת חילוק (div/ldiv).
- המלצה – הגדירו והשתמשו במילון הבא:
Dictionary DB '0', '1', '2', ..., '9', 'A', 'B', ...
- הכניסו ערך כלשהו ל-AX ותוודאו שאכן הודפס הערך הרצוי.
- על התכונות לעבוד לכל ערך של AX – שימו לב להבדל בין ערכי ה-ASCII של המספרים והאותיות.
- קישור לטבלת ASCII - <http://www.asciitable.com>.
- לקובץ הקוד יש לתת את השם: printAXh.asm.

- 2) עליכם להדפיס למסך את התוכן של האוגר AX בייצוג דצימלי (בסיס 10) signed.
לדוגמא עבור $AX = 1F2Ch$ יודפס למסך $AX = 7980d$,
ועבור $AX = F12Ch$ יודפס למסך $AX = -3796d$.

הערות ודגשים:

- הכניסו ערך כלשהו ל-AX ותוודאו שאכן הודפס הערך הרצוי.
- על התכונות לעבוד לכל ערך של AX.
- לקובץ הקוד יש לתת את השם: printAXd.asm.

2. עצרת:

בתרגיל זה עליכם לחשב* ולהדפיס למסך את הערך של הפונקציה הבאה:

$$\sum_{x=n_1}^{n_2} (x+1)!$$

(1) כאשר n_1 הינה הספרה הראשונה בת"ז של סטודנט א, ו- n_2 הינה הספרה האחרונה בת"ז של סטודנט ב (סטודנט א במקרה של הגשה ביחיד). כמו כן, לספרות 8 ו-9 אנו נתייחס כאל הספרה 7.

לדוגמא, עבור $ID_1 = 123456789$, $ID_2 = 444456789$ יתקבל $n_1 = 1$, $n_2 = 7$ ותוצאת החישוב תהיה:

$$\begin{aligned} \sum_{x=1}^7 (x+1)! &= 2! + 3! + 4! + 5! + 6! + 7! + 8! \\ &= 2 + 6 + 24 + 120 + 720 + 5040 + 40320 \\ &= 46232_d = B498_h \end{aligned}$$

כלומר על המסך צריך להופיע $B498_h$.

- על הקוד לעבוד לכל ת"ז.
- לקובץ הקוד יש לתת את השם: factolD.asm.

(2) כאשר $n_1 = 0$ ו- $n_2 = 9$. כלומר:

$$\begin{aligned} \sum_{x=0}^9 (x+1)! &= 1! + 2! + 3! + 4! + 5! + 6! + 7! + 8! + 9! + 10! \\ &= 1 + 2 + 6 + 24 + 120 + 720 + 5040 + 40320 + 362880 + 3628800 \\ &= 4037913_d = 3D9D19_h \end{aligned}$$

לכן, על המסך צריך להופיע $003D9D19_h$ (ניתן גם ללא האפסים המובילים).
• לקובץ הקוד יש לתת את השם: factol09.asm.

(3) בנוס - כאשר $n_1 = 0$ ו- $n_2 = F_h = 15_d$.
• לקובץ הקוד יש לתת את השם: factol0F.asm.

*בתרגיל זה פעולות הסכימה והעצרת נדרשות להתבצע בלולאה. אין להשתמש ב-lookup table. כמו כן, עליכם לראות את ערכי האוגרים בתוכנת הדיבאגר ולצרף צילום מסך להגשה.

הערות ודגשים:

- רמז - חישוב מה ההבדל בתוצאה ובחישוב בין שלושת הסעיפים.
- שימו לב לפקודת הכפל בה אתם משתמשים – mul / imul.
- המלצה – היעזרו בפקודה ADC dest, src אשר מוסיפה את ערך ה-carry מאוגר הדגלים בנוסף לחיבור הרגיל, $dest = dest + src + CF$. מידע נוסף אודות הפקודה ניתן למצוא בעמוד 67 ב-Intel8086 - full spec שבפיאצה או לחפש בגוגל.
- ניתן ומומלץ להיעזר בתרגיל 1 לצורך הדפסת תוכני האוגרים.