

מיקרו מעבדים ושפת אסמבלר תרגיל 3 – תשפ"ב

אחראית תרגיל- מאיה כתר

תאריך הגשה: 22.5.2022

ההגשה בזוגות, אחד מכל זוג מגיש.

נא להוסיף בתחילת כל קובץ הערה עם שמות המגישים ותעודות זהות.

הגישו תיקיית zip בשם Name1_ID1_Name2_ID2.zip הכוללת את החלק היבש ואת קבצי *asmn* הנדרשים בחלק הרטוב.

חלק יבש:

תרגיל 1:

נתונים שני קטעי קוד:

CodeB:

```
mov ax, 5000h
mov bx, FFF0h
test ax, bx
ja L3
```

CodeA:

```
mov ax, 5000h
mov bx, FFF0h
cmp ax,bx
ja L2
```

(1) האם תתבצע קפיצה ל-L2?

(2) האם תתבצע קפיצה ל-L3?

(3) הפקודה `test ax, bx` ב-CodeB הוחלפה בפקודה `cmp bx,ax` (שימו לב לסדר השונה של הרגיסטרים). שנו את הפקודה האחרונה ב-CodeB כך שיבצע אותה פעולה בדיוק כמו CodeA

תרגיל 2:

רשמו דוגמה אחת של ערכי AX,BX שבעזרת פקודת חיבור או חיסור יקיימו את התנאים הרשומים. אם לא ניתן למצוא ערכים המקיימים את הרשום בסעיף, יש לענות "לא קיים"

א. פקודה שתדליק בו זמנית את הדגלים OF,CF,SF

ב. פקודה שתדליק בו זמנית את הדגלים OF,CF,ZF

חלק רטוב:

תרגיל 1:

בתרגיל זה נעסוק באלגוריתם רקורסיבי לחישוב GCD (מחלק משותף מקסימלי).

האלגוריתם למציאת GCD בין שני מספרים חיוביים ושלמים מוגדר כך:

```
recGCD(a,b) :  
  if b==0 then  
    return a  
  else  
    return recGCD(b, a mod b)
```

כאשר $a \bmod b$ היא השארית המתקבלת מחילוק של a ב- b .

(א) ממשו את השגרה `recGCD` לפי האלגוריתם הנתון.

הפרמטרים מועברים לשגרה ע"י אוגרים. קלט השגרה: $AX=a, BX=b$.

פלט השגרה מועבר חזרה ע"י הזיכרון: הגדירו בסגמנט הנתונים משתנה בשם `result` בגודל מילה אשר ישמש להחזרת הפלט.

שימו לב שהשגרה חייבת לשמר את מצב המערכת (אוגרים וזיכרון) מנקודת המבט של התוכנית הקוראת את השגרה.

(ב) כתבו שגרה `arrGCD` המחשבת את GCD של מערך מילים (DW) בשם `input_arr` ושומרת

את התוצאה ברגיסטר `AX`. יש להשתמש בשגרה מסעיף א.

על השגרה להיות רקורסיבית כאשר מתקיים: $\text{gcd}(a,b,c,d)=\text{gcd}(a, \text{gcd}(b,c,d))$

```
.model small  
.data  
  result dw ?  
  input_arr dw 3024, 1234, 1244, 44, 12414  
  inputLen EQU 5  
.code
```

דגשים:

- חובה להוסיף הערות והסברים לקוד.
- שמרו בקובץ בשם `GCD.asm`

שאלות עיוניות:

את התשובות לשאלות אלה יש לצרף יחד עם התשובות לחלק היבש.

(1) חשבו מה גודל המחסנית שנדרש עבור הפעלת השגרה `arrGCD` עבור המערך הבא:

1234	3096	12440	2048
------	------	-------	------

(2) מה סיבוכיות הזמן של התוכנה (כמה פקודות ירוצו, עבור מערך בגודל n)?

(3) הגדירו את `input_arr` להיות מערך גדול (למשל, בגודל 500, או אפילו יותר על מנת לקבל תוצאה מדויקת יותר). בעזרת פקודת הדוס `TIME` נסו למדוד את זמן הריצה של התוכנית עבור חישוב GCD על המערך הגדול. לפי מדידה זו, חשבו מהו משך הזמן הממוצע של פקודה יחידה.

תרגיל 2:

כתבו את השגרה numPrefix המקבלת מספר שלם בפורמט הקסה-דצימלי דרך המחסנית ומדפיסה את השורות הבאות:

בשורה הראשונה יופיע המספר כולו בערכו הדצימלי.

בשורה הבאה – המספר בערכו הדצימלי ללא ספרת האחדות שלו.

בשורה שאחריה – המספר בערכו הדצימלי ללא ספרות העשרות והאחדות, וכך הלאה.

עד שבשורה האחרונה תודפס הספרה המשמעותית ביותר של המספר בערכו הדצימלי.

לדוגמה, עבור קלט: 746FH יודפסו השורות הבאות:

29807

2980

298

29

2

(שכן $746Fh = 29807d$)

שימו לב: שהשגרה חייבת לשמר את מצב המערכת (אוגרים וזיכרון) מנקודת המבט של התוכנית הקוראת את השגרה. כמו כן על השגרה להיות רקורסיבית.

הפעילו את השגרה כאשר המספר המתקבל דרך המחסנית הוא 2 הספרות הראשונות בת"ז של סטודנט א משורשרות ל2 הספרות הראשונות של סטודנט ב.

לדוגמה: עבור $ID_1 = 123456789$, $ID_2 = 333375699$ המספר שנכניס דרך המחסנית יהיה 1233.

- במקרה של הגשה ביחיד ניקח את 4 הספרות הראשונות של התז

הערות ורמזים:

- הקלט הוא מילה הקסדימלית (16 ביט), עליו נרצה לחשוב בתור מספר unsigned
- מומלץ ליצור פונקציית עזר להדפסה למסך.
- אין להשתמש ב-lookup table, אלא לחשב את הערך שצריך להדפיס בתוך הפונקציה numPrefix.
- שמרו בקובץ בשם exPrefix.asm
- חובה להוסיף הערות והסברים לקוד.
- נא להוסיף צילום מסך של הפלט המודפס למסך