

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 ספרות 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

התלויים
הפירו
לקדמות
בסיס התכנית
כפרים

הרבה תלויים
GCD(16,24)=8
4x 0x

בסיס התכנית אלכו קיבלנו 8
טלה של אינדיקס 0800
אזק נמוך

בסוף הרצה קיבלנו ב AX את התוצאה 66

result AX=42h=6d

הרבה טלה ואזכי 2 על תדירות הרבה אלכו 12594d=3132h

C:\B08...DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: COMMAND

File Edit Search View Encoding
Language Settings Tools Macro
Run Plugins Window ?

ex\refix.asm

76 ;L2:
77 ;mov dx,1
78 ;push dx
79 ;loop L2
80
81 ;mov cx,3
82 ;L3:
83 ;mov dx,1
84 ;pop dx
85 ;loop L3
86
87 ;setting ds to d
88 mov ax, 8data
89 mov ds,ax
90
91 mov ax, 0b000b
92 mov es, ax
93
94 mov ax, 3132h ;3
95 ;mov ax, 0015d ;
96 push ax
97 mov cx, 0
98 call numPrefix
99
100 ;return to 00
101 mov ax, 4c00h
102 int 21h
103
104 End START
105 ;-----

FreeDos) C:\MASM\BIN>expref ix.exe
FreeDos) C:\MASM\BIN>_

12594
1259
125
12
1

Windows UTF-8INS
22:39

מיקור סיפורים עמוד 7