תרגיל בית 4 - מיקרו מעבדים ושפת אסמבלר - עידו שר שלום

חלק א' - תרגיל יבש

א. מטרת השגרה VEC_MUL היא סכימת מכפלת איבר איבר בין הוקטורים VEC1 ו- VEC2 והשמת הערך באוגר AX.

ב.

שיטת העברת הפרמטרים לשגרה היא העברת פרמטרים במחסנית.
לשגרה מועברים שלושה פרמטרים:

offset VEC1 - כתובת המשתנה VEC1 בסגמנט הנתונים, ,תפקיד הפרמטר הוא שיוך מצביע למערך בזיכרון המייצג את הוקטור הראשון.

offset VEC2 - כתובת המשתנה VEC2 בסגמנט הנתונים, ,תפקיד הפרמטר הוא שיוך מצביע למערך בזיכרון המייצג את הוקטור השני.

LEN - ערך המשתנה LEN בסגמנט הנתונים, תוכן הפרמטר הוא אורך הוקטורים VEC1 ו - VEC2.

ג. שיטת העברת הפרמטרים חזרה מהשגרה היא העברה דרך אוגר.

מועבר פרמטר יחיד האוגר AX (תוצאת השגרה), תוכן האוגר מציין את תוצאת הסכום של מכפלת איבר איבר בין הוקטורים, התוצאה נשמרת באוגר עם חזרת השגרה לתכנית המזמנת.

ד. a. האוגר CX אינו מאותחל בערך הנכון מכיוון שבקריאה לשגרה מתבצעת דחיפת כתובת החזרה למחסנית (IP במקרה שלנו כי מדובר בשגרת NEAR) וקפיצה לתחילת השגרה.

תוכן המחסנית בקריאה לשגרה:

תוכן המחסנית	הערות
IP	STACK Pointer register points here
LEN	פרמטר
offset VEC2	פרמטר
offset VEC1	פרמטר
zzzzzz	ערכי זבל
zzzzzz	ערכי זבל

ולכן כאשר מתבצע POP ערך האוגר CX הוא ערך כתובת החזרה IP ולא ערך הפרמטר LEN כפי שהיינו רוצים.

.b

השינויים בשגרה:

נתונה השגרה הבאה:

```
0026                VEC_MUL PROC NEAR
0026  59                POP CX
0027  5E                POP SI
0028  5F                POP DI

                        ; Do the work
0029  33 C0             XOR AX, AX
002B  8A 04             MOV AL, [SI]
002D  F6 2D             IMUL BYTE PTR [DI]     ; Multiplication output in AX

                        ; Base Case
002F  83 F9 01          CMP CX,1
0032  75 01             JNZ REC
0034  C3                RET

                        ; Recursion Case
0035                    REC:
0035  50                     PUSH AX     ; save for later

0036  46                     INC SI
0037  47                     INC DI
0038  49                     DEC CX
0039  57                     PUSH DI
003A  56                     PUSH SI
003B  51                     PUSH CX
003C  E8 FFE7                CALL VEC_MUL

003F  5B                     POP BX      ; restore saved AX
0040  03 C3                  ADD AX, BX
0042  C3                     RET
0043                VEC_MUL ENDP
```

mov BP,SP
mov CX,[BP+2] ;get LEN
mov SI,[BP+4] ;get offset VEC2
mov DI,[BP+6] ;get offset VEC1

RET 6

RET 6

הסבר השינויים:

על ידי שימוש במצביע בסיס המחסנית BP ניתן להוציא את כל ערכי הפרמטרים לרגיסטרים המתאימים וכך לדאוג שהאוגר CX לא יקבל ערך שגוי, וכך גם SI ו- DI.

על מנת לחזור מהשגרה בצורה טובה יש לשנות את היסט המצביע לראש המחסנית רגיסטר SP, הפקודה RET 6 דואגת לעשות זאת.

מכיוון ש-בחזרה מהשגרה הערך העדכני של הסכום חייב להיות בראש המחסנית ואם לא נעשה RET 6 יהיו בראש המחסנית ערכי הפרמטרים, מה שיגרום לקריאות רקורסיביות חוזרות לקבל ערך שגוי כל פעם, ומונה הסכום לא מתעדכן לערך הנכון.

ה. נשים לב כי מתבצעת מכפלה של 8 סיביות (byte ב - byte)

DI מצביע בזיכרון לתחילת מערך של בתים על מנת שהמכפלה תתבצע כמכפלת בית ב- בית כלומר הכפלת האיבר הראשון בוקטור הראשון בהכפלת האיבר הראשון בוקטור השני נצטרך להגיד מפורשות למהדר איזו מכפלה מתבצעת (מכפלת בית ב- בית או מכפלת מילה ב- מילה).
אחרת, יכול להיות מצב שבו תהיה מכפלת מילה במילה, שתי הבתים הראשונים ש - DI מצביע עליהם (שתי האיברים הראשונים בוקטור) ו- רגיסטר AX כאשר ב - AH יש ערך זבל שהוא בכלל לא עדכני לערך הוקטור האחר ונקבל תוצאה שגויה.
ולכן במקרה זה נדרש להנחות את המהדר בצורה מפורשת.

חלק ב' - תרגיל רטוב

1. הדפסת המחסנית למסך:

דוגמת הרצה מצורפת.

2. LCS – Longest Common Subsequence:

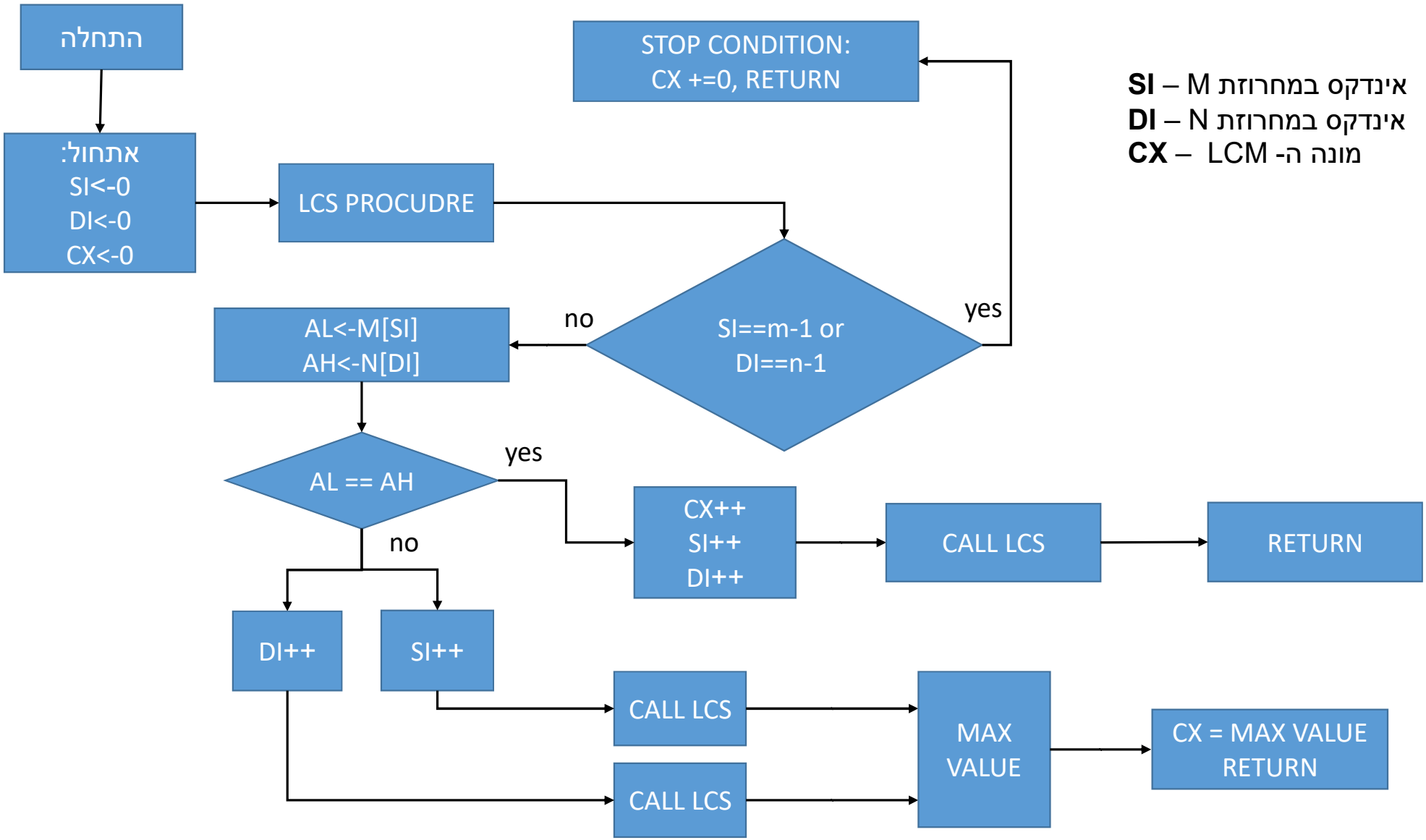
פתרון סעיפים a ו- b בעמוד הבא.

c. המקרה הגרוע ביותר הוא כאשר שתי המחרוזות שונות זו מזו בכל תו ותו במקרה זה, כל פעם תתבצע איטרציה נוספת ועץ הרקורסיה יגדל מה שיגרור לסיום האיטרציות רק בסוף המחרוזות, קרי בתו האחרון.
במצב כזה, זה תלוי בגודל המחרוזות נסמן r = max(n,m),
בכל קריאה לפונקציה אנחנו מכניסים 4 אוגרים si,di,cx,ip בנוסף אוגר עזר ax בשימוש הפרוצדורה.
בהנחה שקיים סגמנט אחד לזיכרון המחסנית 64k כתובות, אורך המחרוזת הגדולה ביותר r תהיה:
64k/4-1 = 65,536/4-1 = 16,383.

d. בהמשך לסעיף הקודם, במקרה הגרוע ביותר עבור כל תו במחרוזת M נצטרך להשוות כל תו עם המחרוזת N מה שגורר סיבוכיות זמן של $o(2^r)$, בכל פעם יש פיצול של שני ענפים רקורסיות בעץ.
סיבוכיות זיכרון, קודם כל יש לנו זיכרון של המחרוזות M ו- N בנוסף זיכרון מחסנית שלא יעלה על 4r במקרה הכי גרוע יהיה רצף של r קריאות למחסנית כאשר שתי המחרוזות שונות בכל איטרציה כזאתי נשמרים על המחסנית 4 רגיסטרים נוספים, סה"כ סיבוכיות זיכרון:

$$O(n + m + 4r) = O(r)$$

ראינו בקורס מבני נתונים ואלגוריתמים כי יש דרכים נוספות לפתרון הבעיה, למשל בעזרת תכנון דינמי נקבל סיבוכיות זמן וזיכרון של o(nm) כנראה שדרך זו היא טובה יותר, אך יותר קשה למימוש בשפת אסמבלר.



SI – אינדקס במחרוזת M
DI – אינדקס במחרוזת N
CX – מונה ה- LCM

דוגמאות הרצה בתרגיל הרטוב:

1. הדפסת המחסנית:

בדוגמה המצורפת, המחסנית היא עם 19 איברים.

```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   DOSBOX
Z:\>mount c: c:\8086                                    14
Drive C is mounted as local directory c:11086\
                                                        12
Z:\>c:                                                  0F
                                                        10
C:\>cd bin                                              0D
                                                        0E
C:\BIN>ml /Zm PrintStk.asm                              0B
Microsoft (R) Macro Assembler Version 6.0C
Copyright (C) Microsoft Corp 1981-1993. 09ll rights reserved.
                                                        0A
 Assembling: PrintStk.asm                               07
                                                        08
Microsoft (R) Segmented Executable Linke06 Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. 05ll rights reserved.
                                                        04
Object Modules [.obj]: PrintStk.obj                     03
Run File [PrintStk.exe]: "PrintStk.exe"                 02
List File [nul.map]: NUL                                01
Libraries [.lib]:
Definitions File [nul.def]:

C:\BIN>PrintStk.exe

C:\BIN>_
```

2. צילום הדיבגר בבעית ה LCM:

```
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip  0, Program:   CV
 File  Edit  Search  Run  Data  Options  Calls  Windows  Help
[1]                                  [2]------ watch ------
                                     The length of the longest common
                                     subsequence
[3]------------------ source1 CS:IP ------------------
05E6:0003 8ED8        MOV     DS,AX
05E6:0005 33F6        XOR     SI,SI
05E6:0007 33FF        XOR     DI,DI
05E6:0009 33C9        XOR     CX,CX
05E6:000B E80700      CALL    0015
05E6:000E 90          NOP
05E6:000F 90          NOP
05E6:0010 B8004C      MOV     AX,4C00
05E6:0013 CD21        INT     21
05E6:0015 3B363300    CMP     SI,WORD PTR [0033]
05E6:0019 7439        JZ      0054
                                     The strings N and M
[5]------------------ memory1 b ds:0006 ------------------
05EB:0006  4D 65 73 73 61 67 65 20 31 20 67 6F 65   Message 1 goe
05EB:0013  73 20 68 65 72 65 21 4D 65 73 73 61 67   s here!Messag
[9]------------------ command ------------------
<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt>

=[7]reg ==
 AX = 05EB
 BX = 0000
 CX = 000F
 DX = 0000
 SP = 1000
 BP = 0000
 SI = 0008
 DI = 0008
 DS = 05EB
 ES = 05D6
 SS = 05EF
 CS = 05E6
 IP = 000F
 FL = 0246

NV UP EI PL
ZR NA PE NC
DEC
```