

המרכז הישראלי לחינוך
מדעי-טכנולוגי
ע"ש עמוס דה-שליט



משרד החינוך
האגף לתכנון ולפיתוח
תכניות לימודים



האוניברסיטה הפתוחה
בית הספר לטכנולוגיה



המרכז לטכנולוגיה
חינוכית (מט"ח)



CMP BX, 0	
JG 000B	
MOV AX, 1	
JMP 0013	
PUSH BX	
DEC BX	
CALL 0000	
POP BX	
IMUL BX	
RET	

מקטע מחסנית

00FE	001F
00FC	0003
00FA	0010
00F8	0002
00F6	0010
00F4	0001
00F2	0010
00F0	



מבוא למערכות מחשב ואסמבלי

פעילויות לתלמיד

משרד החינוך האגף לתכנון ולפיתוח תכניות לימודים
המרכז לטכנולוגיה חינוכית (מט"ח)
המרכז הישראלי לחינוך מדעי – טכנולוגי ע"ש עמוס דה שליט
האוניברסיטה הפתוחה בית הספר לטכנולוגיה

מבוא למערכת מחשב ואסמבלי

פעילויות לתלמיד

כתיבה

שרה פולק
דינה קראוס

קריאה וייעוץ מדעי-פדגוגי

ד"ר צבי פירסט
ישראל זילברשטיין
עריכה לשונית
יעל מילר

הפקה

אביבה אבירן

עיצוב עטיפה

אבי חתם

מקט 1043312

מהדורת ניסוי

© מהדורת תשס"ז - 2006. כל הזכויות שמורות למשרד החינוך.
בית ההוצאה לאור של המרכז לטכנולוגיה חינוכית, קרית משה רואו, רח' קלאוזנר 16,
רמת-אביב, ת"ד 39513, תל"אביב 61394.

The Center For Educational Technology, 16 Klausner St., Ramat-Aviv,
P.O.Box 39513, Tel-Aviv, 61394. Printed in Israel

זכויות הקניין הרוחני, לרבות זכויות היוצרים והזכות המוסרית של היוצר/ים בספר
זה מוגנות. אין לשכפל, להעתיק, לסכם, לצלם, להקליט, לתרגם, לאחסן במאגר
מידע, לשדר או לקלוט בכל דרך או בכל אמצעי אלקטרוני, אופטי, מכני או אחר,
כל חלק שהוא מספר זה. כמו כן, אין לעשות שימוש מסחרי כלשהו בספר זה, בכולו
או בחלקים ממנו, אלא אך ורק לאחר קבלת רשות מפורשת בכתב ממטח(המרכז
לטכנולוגיה חינוכית).

תוכן עניינים

פעילות 1 - מבנה המחשב ולשפת מכונה	5
פעילות 2 - המודל התכנותי של מעבד 8086	11
פעילות 3 - תהליך כתיבת תכנית בשפת סף והרצתה	15
פעילות 4 - שימוש בהוראות העברה ובהוראות חשבון לחיבור ולחיסור	31
פעילות 5 - אוגר הדגלים והוראות קפיצה	73
פעילות 6 - לולאות	94
פעילות 7 - הוראות לוגיות	55
פעילות 8 - הוראות הזזה והוראות סיבוב	59
פעילות 9 - שיטות מיעון ומערכים	67
פעילות 10 - פרוצדורות ומקרו	73
פעילות 11 - קלט פלט ופסיקות	81
פעילות 12 - מחרוזות	83
פעילות 13 - מעבדים מתקדמים	91

מבנה המחשב ולשפת מכונה

חומר רקע

פעילות זו מתייחסת לפרק 1 בספר הלימוד. בפעילות זו נשתמש בהוראות המכונה של המחשב הפשוט שהוצגו בפרק הראשון בספר הלימוד והן מוצגות בטבלה הבאה.

טבלה 1

סיכום כל הוראות המכונה של המחשב הפשוט

קוד פעולה	תיאור ההוראה	סוג הוראה
00	סיום	
01	נתון אוגר יעד	העברה
02	אוגר מקור אוגר יעד	
03	תוכן כתובת בזיכרון אוגר יעד	
04	נתון כתובת בזיכרון	
05	אוגר מקור כתובת בזיכרון	
06	נתון + אוגר אוגר	חיבור
07	אוגר מקור + אוגר יעד אוגר יעד	
08	האם הנתון = לאוגר יעד?	השוואה
09	האם האוגר מקור = לאוגר יעד?	
10	קפיצה מותנית	קפיצות
11	קפיצה בלתי מותנית	



תרגיל 1

הסבירו את המושגים הבאים:

- א. תוכנה
- ב. חומרה
- ג. שפת מכונה
- ד. שפת אסמבלי

תרגיל 2

ציינו אם הטענות להלן נכונות או לא ונמקו:

- א. תכניות שנכתבו בשפות תכנות נמוכות יכולות להתבצע על כל מחשב.
- ב. תכניות שנכתבו בשפות תכנות עיליות לאחר הידור יכולות להתבצע על כל מחשב.

תרגיל 3

הסבירו את המושגים הבאים:

- א. אסמבלר
- ב. מהדר
- ג. מפרש

תרגיל 4

הסבירו מהי מערכת הפעלה ומהם תפקידיה.

תרגיל 5

הסבירו את המושגים:

- א. ארכיטקטורה של מחשב.
- ב. ציינו מהם המרכיבים של ארכיטקטורה של מחשב והסבירו מה הם מתארים.

תרגיל 6

הסבירו מה היה החידוש העיקרי בארכיטקטורת פון ניומן.

תרגיל 7

הסבירו את המושג מחזור הבאה-ביצוע.

תרגיל 8

הסבירו מדוע במחשב הפשוט המתואר בפרק הראשון בספר יש לדאוג להפרדה בין הוראות התכנית לבין הנתונים שלה.

תרגיל 9

ציינו מהם האופרטורים ומהם האופרנדים בהוראה

$$a = a * (c+b)$$

תרגיל 10

האם הוראת המכונה (של המחשב הפשוט) הבאה

04 90 80

מבצעת את הפעולה "החלף בין תוכן כתובת תא 90 בתוכן כתובת תא 80"?
אם לא, ציינו מהי הטעות וכתבו הוראות מתאימות.

תרגיל 11

הסבירו מה מציין המספר 06 בהוראות המכונה (של המחשב הפשוט) הבאות:

א. 06 02 01

ב. 04 06 02

ג. 03 02 06

תרגיל 12

להלן תכנית בשפת מכונה של המחשב הפשוט

04 97 10
04 98 12
04 99 20
03 01 97
03 02 98
07 01 02
03 02 99
07 01 02
00

א. הסבירו מה מבצעת התכנית.

ב. להלן מפת זיכרון לאחר ביצוע ההוראה השישית: 07 01 02
השלימו את הערכים החסרים.

זיכרון		מעבד	
		יחידת ביצוע	יחידת בקרה
0	04 97 10	DR1	IR 07 01 02
1	04 98 12	DR2	IP
2	04 99 20		
3	03 01 97		
4	03 02 98		
5	07 01 02		
6	03 02 99		
7	07 01 02		
8	00		
	.		
	.		
97			
98			
99			

תרגיל 13

א. הגדירו סוג חדש של הוראה שתבצע את הפעולה הזו:

1 - אופרנד -> אופרנד

כאשר סוגי האופרנדים האפשריים הם: אוגר DR1 או אוגר DR2

ב. השתמשו בהוראות שהגדרתם ובהוראות שתוארו בסעיף זה כדי לבצע את הפעולות הבאות:

```
DR1 ← 5
DR2 ← 0
כל עוד  $DR1 > 0$  בצע
 $DR1 ← DR1 + DR2$ 
 $DR1 ← DR1 - 1$ 
```

תרגיל 14

א. הסבירו את המושג מחזור פס.

ב. תארו מחזור פס לביצוע שלב ההבאה ומחזור פס לביצוע שלב הביצוע של ההוראה 10 98 04.

תרגיל 15

נניח כי במחשב שלנו חיברו את יחידת הקלט לכתובת 99 ואת יחידת הפלט לכתובת 98. כתוצאה מכך תחום הכתובות של תאי הזיכרון הוקטן וכעת הוא בין 0 ל-97.

א. תארו מחזור פס שבו קורא המעבד מיחידת קלט את הנתון 90 אל אוגר DR1.

ב. תארו מחזור פס שבו כותב המעבד את תוכן אוגר DR1 ביחידת הפלט.

המודל התכנותי של מעבד 8086

חומר רקע

פעילות זו מתייחסת לפרק 4 סעיפים 4.1-4.3 בספר הלימוד



תרגיל 1

רשמו באילו אוגרים אפשר לאחסן:

- א. נתונים בלבד
- ב. כתובות בלבד
- ג. נתונים וכתובות

תרגיל 2

האם הטענה הבאה היא נכונה:

לכל האוגרים במעבד 8086 אפשר להתייחס כאל אוגרים בני שמונה סיביות ולפנות או לבית העליון או לבית התחתון. הסבירו מדוע.

תרגיל 3

המירו את הגדלים הבאים מיחידת מידה אחת ליחידת מידה שנייה:

- א. $2048K = M$?
- ב. $K = 1T$?
- ג. $G = 4096E$?

תרגיל 4

הסבירו כיצד שיטת מיעון מקטעים מבטיחה כי תכניות שנכתבו למעבד 8085 יוכלו להתבצע גם במעבד 8086?

תרגיל 5

את מרחב הזיכרון של מעבד 8086 חלקו למקטעים רצופים שאינם חופפים. גודל כל אחד מהם הוא 64KB מהי הכתובת הפיזית של התא הראשון במקטע החמישי? מהי כתובתו היחסית של אותו תא?

שאלה 6

הגדירו שני מקטעים בגודל 64KB. המקטע הראשון מתחיל בפסקה 0 והמקטע השני מתחיל בפסקה 1. האם הטענות להלן נכונות? נמקו:

- א. שני המקטעים אינם חופפים זה לזה כלל.
- ב. קיימת חפיפה מלאה בין שני המקטעים.
- ג. לפחות 80% מהמקטע הראשון חופף למקטע השני.

תרגיל 7

הסבירו את יתרונות השימוש בשיטת המקטעים לעומת השימוש בזיכרון הליניארי. כיצד המבנה של מעבד 8086 תומך בשיטת המקטעים לארגון הזיכרון?

תרגיל 8

הגדירו את המשתנים הבאים במקטע הנתונים:

.DATA

a	DB	10h
b	DW	0DoFFh
c	DW	12AFh

למקטע הנתונים הוקצה מקטע המתחיל בכתובת $DS = 1000h$ רשמו את הכתובת היחסית ואת הכתובת הפיזית של המשתנים.

תרגיל 9

במקטע נתונים הוגדרו המשתנים הבאים:

.DATA

x	DB
y	DT
q	WD
t	DQ

- א. חשבו מהו גודל מקטע הנתונים הדרוש.
- ב. מהי הכתובת היחסית של המשתנה האחרון?
- ג. האם אפשר לחשב את הכתובת המוחלטת של המשתנה האחרון לפני טעינת התכנית לזיכרון? אם כן, הסבירו כיצד. אם לא אפשר לחשב כתובת זו, הסבירו מדוע.

תהליך כתיבת תכנית בשפת סף והרצתה

חומר רקע

פעילות זו מתייחסת לפרק 4 סעיף 4.3 בספר הלימוד שבו מוצגים הנושאים הבאים:
תהליך הרצת תכנית אסמבלי, הידור, אסמבלר, הנחיות אסמבלר, מבנה תכנית, קישור, שימוש ב- debugger.

תהליך הרצת תכנית

תהליך פיתוח של תכנית כולל מספר שלבים כגון: פירוק בעיה לבעיות משנה, הגדרת משתנים, תיאור אלגוריתמים ולסיום, מלמוש בשפת תכנות, הרצה ובדיקה. בפעילות זו נתמקד בשלב ההרצה של תכנית בשפת אסמבלי ונכיר את כלי ההרצה: האסמבלר והדיבגר.

תהליך ההרצה של תכנית בשפת אסמבלי מורכב ממספר שלבים המתוארים באיור 1. המלבנים באיור מתארים את האמצעים שבהם משתמש המתכנת בכל שלב והעיגולים מתארים את סוג הקובץ המופק.

בשלב הראשון אנו כותבים את התכנית בשפת אסמבלי בעזרת **תכנית עריכה** (editor) כגון:

Edit או Notepad. להוראות תכנית בשפת אסמבלי יש מבנה קבוע שאותו נתאר בהמשך. הקובץ הנוצר בשלב זה נקרא **קובץ תכנית המקור** (source program file) או בקיצור **קובץ מקור** (source file) והסיומת שלו היא ASM.

בשלב השני עלינו לתרגם את תכנית המקור לשפת מכונה. לשלב זה אנו קוראים **הידור** ובסיומו מתקבל **קובץ מטרה** (object file) שהסיומת שלו היא OBJ. כדי לבצע

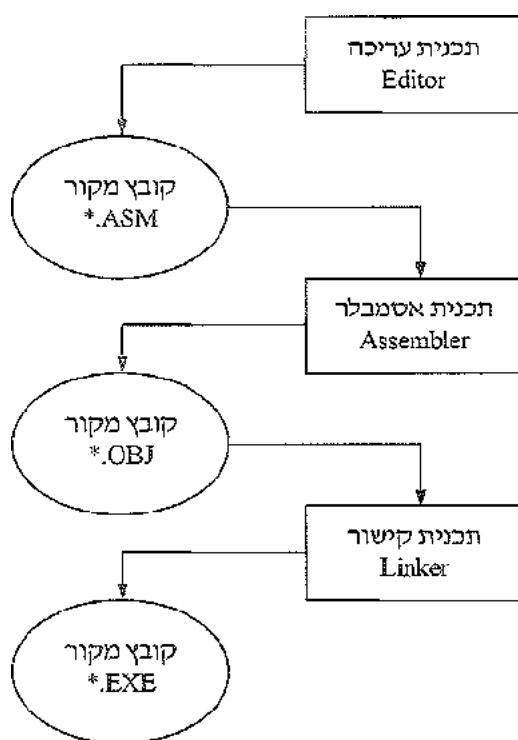
שלב זה אנו משתמשים בתכנית **אסמבלר** (assembler) הקוראת את קובץ המקור ומתרגמת כל הוראה בשפת אסמבלי להוראה בשפת מכונה. לדוגמה, במעבד 86 X 80 ההוראה: `mov ax, 0` מתורגמת לקוד B80000.

בשלב השלישי קובץ המטרה עובר תהליך הנקרא קישור (link) ובסיומו יתקבל קובץ מוכן להרצה (הסיומת שלו היא EXE). תהליך הקישור מבוצע באמצעות תכנית קישור (linker) אשר קוראת את קובץ המטרה ומפיקה ממנו קובץ ריצה (run file). קובץ הריצה מכיל את התכנית בשפת המכונה המוכנה לריצה. קובץ מסוג זה נקרא גם קובץ ביצועי (executable).

נציין כי לכל סוג מעבד יש שפת מכונה שונה. לכן, קובץ המוכן לריצה במעבד מסוים יוכל לרוץ רק עליו. כדי להריצו במעבד אחר נצטרך לבצע שוב את תהליך ההידור והקישור.

למעבדים ממשפחת 80X86 ישנן מספר תוכנות אסמבלר ותוכנות קישור והנפוצות ביותר הן:

MASM (Microsoft's macro assembler) ו-TSAM (Borland Turbo Assembler). בספר זה נתייחס לתוכנת TASM המורצת תחת DOS.



איור 1

תהליך פיתוח תכנית בשפת-סף



תרגיל 1

תיאור המשימה: כתיבת תכנית וביצוע תהליך ההרצה

במשימה זו עליכם להריץ תכנית בשפת אסמבלי. התכנית מציגה את הודעה:
"Hello world , I'm learning Assembly !!!" על הצג.
בתכנית זו מוגדר משתנה בשם message מטיפוס בית והוא מאותחל להודעה
הדרושה.

להלן התכנית הדרושה:

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

הגדרת משתנה המכיל את ההודעה שיש להציג;

```
Message DB "Hello world , I'm learning Assembly !!! ", "$"
```

```
.CODE
```

```
start:
```

אתחול קטע זיכרון המכיל את ההודעה;

```
mov ax, seg message
```

```
mov ds, ax
```

הוראות להצגת ההודעה על הצג;

```
mov ah, 09
```

```
lea dx, message
```

```
int 21h
```

הוראות לסיום תכנית ולהחזרת הבקרה למערכת ההפעלה;

```
mov ax, 4c00h
```

```
int 21h
```

```
END start
```

הגדרת המשימה:

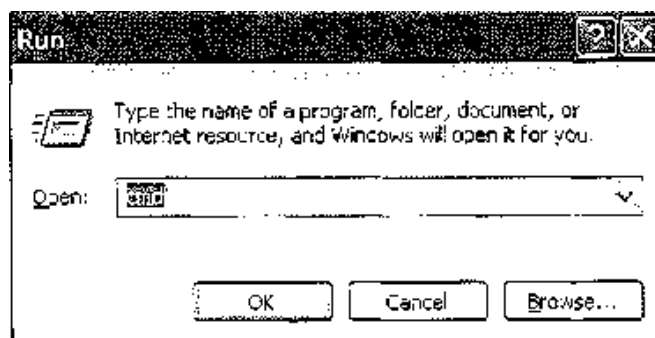
כתבו את התכנית באמצעות עורך והריצו אותה.

ביצוע המשימה:

1. מעבר לחלון DOS

בחרו בתפריט התחל – < הרץ

בחלון שנפתח הקלידו cmd ולסיום הקליקו OK.



ייפתח חלון DOS. עברו למדריך c:>tasm שבו מאוסנת תכנית TASM. שימו לב, עליכם לבדוק היכן מאוסנים האסמבלר והדיבגר במחשבים שלכם ולפעול בהתאם למדריך זה.

2. כתיבת התכנית

פתחו את העורך Notepad (או עורך טקסט אחר) והקלידו את התכנית. בסיום התכנית שמרו את הקובץ ותנו לו את השם:

hello.asm

3. הידור

כדי להדר את התכנית שכתבתם רשמו את ההוראה:

tasm/ zi/ I hello.asm

ההוראה מכילה ארבעה מרכיבים:

- tasm – זו תכנית ההידור (אסמבלר).
- /zi – זהו פרמטר הנחוץ לעבודה עם Turbo – debbuger (זו תוכנה המשמשת להרצה ולמעקב אחר ביצוע התכנית).

- /I - והו פרמטר אופציונלי המורה לאסמבלר ליצור קובץ .lst. קובץ זה מכיל את תכנית המקור, את התכנית המהודרת ופרטים על טבלת הסמלים.
- Hello.asm - זו תכנית המקור שיש להדר. אפשר להשמיט מכתובתה את הסיומת asm ולרשום רק hello. תכנית TASM מתייחסת לקבצים עם סיומת asm כברירת מחדל.

אם לא התקבלה כל שגיאת הידור נקבל הודעה הדומה להודעה הבאה:

```
Turbo Assembler Version 3.2 Copyright © 1988, 1992 Borland International
Assembling file:      hello.asm
Error messages:      None
Warning messages:    None
Passes:              1
```

כתוצאה מביצוע ההידור נקבל קובץ המכיל קוד מכונה בשם:

hello.obj

אם התקבלה הודעת שגיאה כלשהי (Error message), חזרו ותקנו את התכנית שכתבתם בעזרת העורך. אם קיימת שגיאת הידור, לא נוכל לעבור לשלב הבא לפני תיקונה.

4. קישור

כדי לבצע קישור נרשום:

tlink/ v hello

ההוראה מכילה שלשה מרכיבים:

- tlnk - זו תכנית הקישור.
- /v - זהו פרמטר הנחוץ לעבודה עם Turbo- debugger
- Hello - זו התכנית (עם הסיומת obj) שהידרנו בשלב הקודם.

אם לא התקבלה כל שגיאת קישור, נקבל הודעה הדומה להודעה הבאה:

```
Turbo Link Version 3.2 Copyright © 1988, 1992 Borland International
```

כעת קיבלנו קובץ מוכן להרצה שהסיומת שלו:

hello.exe

5. הרצה

כדי להריץ את התכנית יש לרשום:

hello

התכנית תתבצע ונקבל את ההודעה:

Hello world, I'm learning Assembly !!!

תרגיל 2

תיאור המשימה: ביצוע שינוי בתכנית

שנו את התכנית שהוצגה במשימה 1 כך, שתציג את ההודעה הבאה:

“Assembly is low-level programming language”

לאחר ביצוע השינוי הדרו את התכנית והריצו אותה.

שלבי הביצוע:

1. השתמשו בעורך ופתחו את התכנית asm.hello.
2. שנו את השורה שבה מוגדר המשתנה message ורשמו:
Message DB “Assembly is low-level programming language”, “\$”
3. שמרו את התכנית בשם אחר. הקפידו להוסיף את הסיומת asm.
4. הדרו את התכנית.
5. בצעו קישור.
6. הריצו את התכנית ובדקו את הפלט שהתקבל.

תרגיל 3

תיאור המשימה: מעקב אחר ביצוע תכנית באמצעות דיבגר

במשימה זו נכתוב תכנית בהתאם לאלגוריתם הבא:

שים 2314 באוגר AX

שים h5061 באוגר BX

חשב $AX = AX + h100$

חשב $AX + BX = BX$

שימו לב: הסימול h בסוף המספר מציין שהמספר מיוצג בשיטה הקסדצימלית. במקרה כזה, אם המספר מתחיל באות, יש להוסיף את הספרה 0 בספרה משמעותית ביותר. לדוגמה: את המספר A70Ch, נרשום כך: 0A70Ch. תחילה נציג את התכנית:

```
.SMALL MODEL
.STACK 100h
.DATA
    a DW 3214h
    b DW 5061h
.CODE
start:
```

אתחול מקטע נתונים;

```
    mov ax, @DATA
    mov ds, ax
    mov ax, a                ; ax = 3214h
    mov bx, b                ; bx = 5061h
    add ax, 100h             ; ax = ax + 100h
    add bx, ax                ; bx = bx + ax
                                ; סיום התכנית;
```

```
    mov ax, 4C00h
    int 21h
```

END start

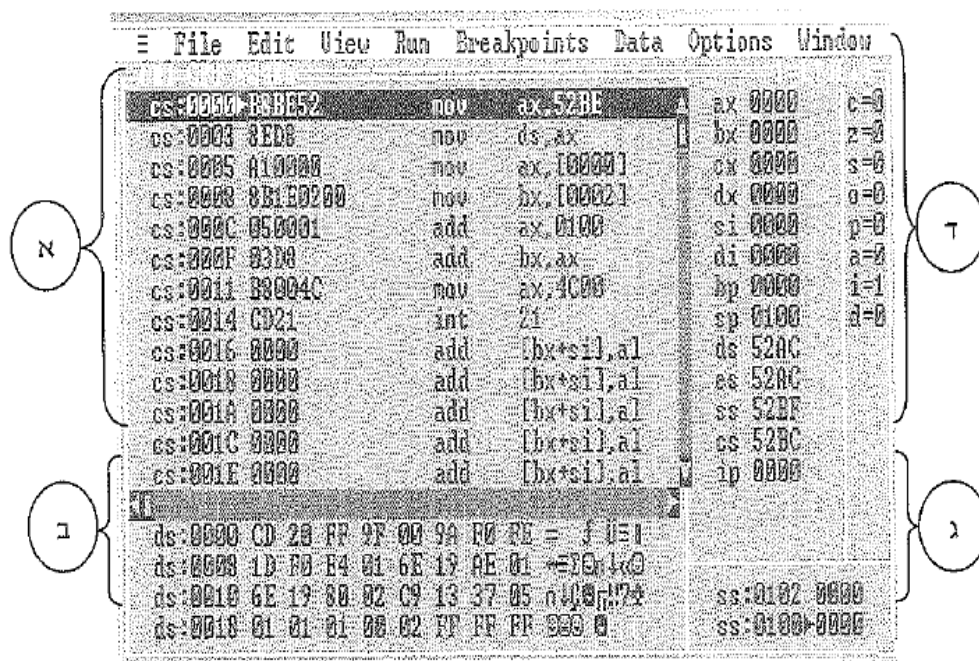
1. כתבו את התכנית ושמרו אותה (הקפידו להוסיף את הסיומת .asm).
2. הדרו את התכנית וקשרו אותה.
3. הריצו את התכנית.
בתכנית זו אין הוראות המציגות פלט על הצג ולכן, כדי לעקוב אחרי מהלך ביצוע התכנית, נשתמש בתוכנת דיבגר (debugger) ששמה TD המצורפת ל - TASM. הדיבגר היא תוכנה המציגה את מפת הזיכרון של המחשב ומאפשרת במהלך הרצת התכנית לעקוב אחרי תוכן הזיכרון.

4. להפעלת תוכנת הדיבגר td ולהרצת התכנית, רשמו:

td test1

החלון שהתקבל מכיל את התכנית שכתבנו. כדי לעקוב אחרי ביצוע התכנית נפתח את חלון ה-CPU.

לשם כך בחרו בתפריט CPU ► View המתואר באיור שלהלן:



איור 2

מסך דיבגר

5. היכרות עם מסך הדיבגר TD:

הדיבגר מציג את תוכן הזיכרון בשיטה ההקסדצימלית ומחולק לארבעה חלקים עיקריים המסומנים באותיות א' – ד':

א. בחלק המסומן ב-'א' מוצגות הוראות התכנית וכל שורה בו מכילה מידע הכולל:
CS: XXXX וקוד מכונה

השדה הראשון CS : XXXX מציין את הכתובת בזיכרון שאליה נטענה הוראה. לכתובת זו יש שני מרכיבים:

- אוגר CS המכיל את הכתובת של תחילת מקטע הקוד שאליה נטענה התכנית בשלב הקישור.
- היסט XXXX הנשמר באוגר IP ומציין את המיקום היחסי של ההוראה במקטע הקוד.

לדוגמה, בשורה הראשונה קיבלנו את המידע הבא:

```
CS:0000    B8BE52    mov ax, 52BE
```

הוראה זו היא המרה של @DATA, ax, mov לאחר הידור וקישור והוחלפה בה התווית

@DATA בכתובת 52BE שהוקצתה למקטע הנתונים

השורה השנייה מכילה את ההוראה השנייה:

```
Cs:0003    8BD8    mov ds, ax
```

ההוראה mov ds, ax שמה את כתובת מקטע הנתונים באוגר הנתונים. הוראה זו מתחילה בהיסט 03 במקטע הנתונים שהיא הכתובת הפיזית $52C6h + 0003H = 52C9h$

השורה השלישית מכילה את התרגום של ההוראה mov ax, a:

```
CS: 0005    A10000    mov ax [0000]
```

אפשר לראות כי לאחר הידור וקישור הוחלף המשתנה a בכתובת יחסית שהוקצתה לו במקטע הנתונים.

כתבו את השורה שהתקבלה עבור ההוראות:

```
mov bx, b
add ax, 100h
add bx, ax
```

ב. בחלק השני כל שורה מציגה מידע המתאר שמונה בתים בזיכרון:

```
ds:XXXX XX XX XX XX XX XX XX zzzzzzzz
```

השדה הראשון מתאר את כתובת הבית הראשון בזיכרון ואחריו מתוארים תוכן שמונת הבתים הראשונים. השדה האחרון מייצג את תוכן שמונת הבתים בקוד אסקי.

ג. בחלק השלישי מוצג מקטע המחסנית. במשימה שלנו לא נשתמש בחלק השלישי ולכן לא נתאר אותו.

ד. בחלק הרביעי מוצגים האוגרים. לאחר ביצוע כל הוראה, מוצג מצבם המעודכן של האוגרים. רשמו בטבלה שלהלן את תוכנם של האוגרים שאתם נעבוד בתכנית זו.

טבלה 2

מעקב אחרי תכנית המוצגת במשימה 3

הוראה	IP	AX	BX	CS	DS
מצב התחלתי					
@DATA mov, ax					
mov ds, ax					
mov ax, a					
mov bx, b					
add ax, 100h					
add bx, ax					
mov ax, 4C00h					

6. הרצת התכנית צעד אחר צעד:
הדיבגר כולל שני סרגלי כלים:

סרגל כלים בשורה העליונה המאפשר פעולות לפתיחה ולסגירה של קובצי הרצה (בעלי סיומת EXE) ולקביעת אופציות שונות של החלונות המוצגים.

סרגל כלים בשורה התחתונה המאפשר בחירה של אופן ההרצה: מקש F7 מורה על הרצת התכנית צעד אחר צעד ומקש F8 מורה על הרצה רציפה. ההוראה העומדת להתבצע מסומנת בצבע לבן.
כל אוגר שיושפע מביצוע הוראה יואר גם הוא.

- לביצוע ההוראה הראשונה לחצו על F7. התבוננו באוגרים שהשתנו ורשמו את תוכנם החדש.
- לביצוע ההוראה השנייה לחצו על F7. התבוננו באוגרים שהשתנו ורשמו את תוכנם החדש.

ג. שימו לב! לאחר ביצוע ההוראה השנייה יוצג מקטע הנתונים הנוסף ES. להצגת

מקטע הנתונים DS, בחרו בתפריט

View dump

את החלון שנפתח אפשר לגרור באמצעות העכבר לכל מקום במסך.

ד. חזרו על שלב א' ובצעו צעד אחר צעד את ההוראות. רשמו בטבלה 1.1 את

תוכנם של האוגרים בכל שלב ושלב.

תרגיל 4

תיאור המשימה: כתיבת תכנית

כתבו תכנית המבצעת את האלגוריתם הבא:

	150h	AX
AX		AX + 2
AX		AX · 2
	BX	AX
	AX	0

ביצוע המשימה

1. כתבו את התכנית.

2. הדרו את התכנית וקשרו אותה.

3. הריצו את התכנית באמצעות הדיבגר צעד אחר צעד. רשמו בטבלת מעקב את

תוכנם של האוגרים:

AX , BX ו- IP בכל שלב ושלב של התכנית.

תרגיל 5

תיאור המשימה: הצגת מספרים בשיטות ספירה שונות

במשימה זו נבחן כיצד מיוצגים תווים ונתונים מספריים הן בשיטה העשרונית והן בשיטה הבינארית.

הטבלה שלהלן מציגה את השיטה לרישום נתונים מסוגים שונים בשפת אסמבלי.

טיפוס	סימול	דוגמאות
תו	בין גרשיים	'a' , 'B' , '3'
מספר עשרוני	קיימות שתי אפשרויות: • ללא סימול מיוחד • הוספת התו d בסוף המספר	9301 , 1203 9301d, 1203d או
מספר הקסדצימלי	הוספת התו h בסוף מספר במספר שמתחיל באות (A – F) הוספת o בספרה השמאלית ביותר	901Dh , oA0d5h , oDFh, gh
מספר בינארי	הוספת התו b בסוף המספר	10010001b , 0001b, 11100000b

כמובן שעלינו לדאוג לרשום נתונים תקינים לפי שיטת הייצוג. אם נרשום מספר שגוי תוכנת אסמבלר תפיק הודעת שגיאה.

שלבי ביצוע

1. רשמו התכנית הבאה :

ייצוג מספרים בשיטה העשרונית ובשיטה הבינארית – asm.test3 ;

```
.model small
.stack 100h
.data
.code
start:
    mov ax, @DATA
    mov ds, ax
```

ייצוג מספרים בשיטה העשרונית;

```
mov ax, 8980d           ; 1) ax = 2341h
mov bx, ax              ; 2) bx = ax
mov bx, 20577d          ; 3) bx = 5061h
add ax, 256d            ; 4) ax = ax + 1000h
add bx, ax              ; 5) bx = bx + ax
                        ; ייצוג מספרים בשיטה הבינארית;
mov ax, 0010001100010100b ; 6) ax = 2341h
```


mov bx, ax	; 7) bx = ax
mov bx, 0101000001100001b	; 8) bx = 5061h
add ax, 0000000100000000b	; 9) ax = ax + 1000h
add bx, ax	; 10) bx = bx + ax

ייצוג מספרים בקוד אסקיי;

mov ax, 'a'	; 11) ax = 'a'
mov bx, 'b'	; 12) bx = 'b'
mov ax, bx	; 13) ax = bx

סיום התכנית;

```
mov ax, 4ch
int 21h
```

END start

2. שמרו את התכנית כקובץ ששמו: test2.asm
3. הדרו ובצעו קישור לקובץ זה. היעזרו במידת הצורך בהנחיות שתוארו במשימה 1.
4. השתמשו בדיבגר והריצו את התכנית צעד אחר צעד. לאחר ביצוע כל הוראה רשמו בטבלת מעקב את תוכנם של האוגרים AX ו-BX.
5. כיצד מוצגים נתונים מסוגים שונים בדיבגר? מדוע לדעתכם משתמשים בדיבגר בשיטת ייצוג זו?

תרגיל 6

תיאור המשימה: תיקון שגיאות דקדוק (syntax error)

במשימה נציג תכנית שגויה ונתייחס להודעות שתוכנת אסמבלר תפיק. התכנית תכיל שגיאות בכתיבת הנחיות אסמבלר ובהוראות עצמן.

שלבי ביצוע

1. רשמו את התכנית הבאה:

```
.MODEL SMALL
.STACK 100h
.DATA
.CODE
```

```

start:
mov ax, @DATA
mov dr, ax
mov ah, 0AF
mov al, FBh
mov bx, al
add cl, 0AF10h
mov al 10
mov ax, 04cooh ; terminate program
int 21h
END start

```

2. שמרו את התכנית בשם: testerror.asm
3. הדרו את התכנית ורשמו:

Tasm testerror

בתגובה תקבלו הצגה של שש הודעות שגיאה.

4. כדי לעקוב אחרי הודעות השגיאה הדרו את התכנית שוב, אך הפעם הפיקו גם קובץ עם סיומת lst. רשמו את הפקודה:

Tasm /1 testerror

השדה 1/ יורה למהדר להפיק (בנוסף לקובץ obj), קובץ list (שהסיומת שלו היא lst). קובץ זה מכליל את תכנית המקור ואת התרגום שלה לשפת מכונה בשיטה הקסדצימלית. כמו כן הוא מכיל מידע על הודעות שגיאה (אם יש שגיאה) ותיאור של מבנה התכנית.

5. פתחו את הקובץ lst.testerror באמצעות עורך והתבוננו בשגיאות שהתקבלו. כל שגיאה שהתקבלה מתוארת לפי הסוג שלה. לדוגמה:

```

70003 A3 0000 mov dr, ax
**Error** testerror.ASM (7) Undefined symbol: DR

```

בהוראה זו השגיאה היא ברישום dr בשם אוגר.

6. לתיקון התכנית, סגרו את קובץ list ופתחו באמצעות העורך את קובץ המקור.
7. לאחר תיקון כל ההוראות, הדרו ובצעו קישור לתכנית.
8. הריצו את התכנית באמצעות הדיבגר ועקבו אחר ביצוע ההוראות.

סיכום

להלן שלד של מבנה תכנית מקור במודל SMALL:

```
model small.  
stack.  
data.  
code.
```

הוראות התכנית
End

להלן תקציר ההוראות לכתיבה ולהרצת תכניות בשפת סף של TURBO ASSEMBLER במערכת הפעלה WINDOWS:

1. בשולחן עבודה בחרו: התחל (START) <— הרץ (Run) ובחלון שנפתח רשמו:
.cmd
2. עברו לספרייה שבה נמצאים קבצי התוכנה. הקבצים הדרושים להרצת תכנית בשפת סף הם:

TASM.EXE TLINK.EXE TD.EXE DPMILOAD.EXE DPMIMEM.DLL

3. לעריכה, רשמו EDIT ולחצו ENTER, או השתמשו בפנקס רשימות (Notepad).
4. כתבו את התכנית ובסיום שמרו אותה עם סיומת: asm.
5. להידור רשמו:
C:> TASM שם תכנית
אין חובה לרשום את הסיומת asm.
6. אם יש טעויות דקדוק חזרו לשלב 3.
7. אם אין טעויות דקדוק בצעו קישור ורשמו:
C:> TLINK שם תכנית
להרצה רשמו:
C:> שם תכנית
או השתמשו בדיבגר ורשמו:
C <:TD שם תכנית

שימוש בהוראות העברה ובהוראות חשבון לחיבור ולחיסור

חומר רקע

פעילות זו מתייחסת לסעיפים 5.1 ו-5.2 בספר הלימוד כדי לעקוב אחרי ביצוע קטע של תכנית אפשר להשתמש בטבלת מעקב שבה אנו מציגים את ערכי המשתנים ואת ערכי האוגרים שבחם אנו משתמשים בקטע הנבדק. לדוגמה, נעקוב אחרי ביצוע קטע מהתכנית שהוצגה בפרק 5 דוגמה 5.1 :

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

```
a DB 38 ; a = 26h
```

```
b DB 23 ; b = 17h
```

```
c DB 36 ; c = 24h
```

```
.CODE
```

```
start :
```

אתחול אוגר סגמנט הנתונים;

```
mov ax, @DATA
```

```
mov ds, ax
```

```
; a = a+b-c
```

```
mov al, a
```

```
; al = 26h
```

```
add al, b
```

```
; al = al + b = 26h + 17h = 3Dh
```

```
sub al, c
```

```
; al = al - c = 3Dh - 24h = 19h
```

```
mov a, al
```

```
mov ax, 4Cooh
```

```
; a = 19h
```

```
int 21h
```

; סיום התכנית

```
END start
```

בטבלה שנבנה נשתמש בחמש עמודות. בעמודה השמאלית ביותר נרשום את ההוראות וארבע העמודות הנותרות יתארו את ערכם של המשתנים a, b, c ואת ערכו של האוגר a1 שבהם נשתמש בתכנית. קטע הקוד בתכנית יכיל הוראות לאתחול מקטע נתונים ולסיום התכנית שאליהן לא נתייחס בטבלת המעקב.

בשורה הראשונה בטבלת המעקב נרשום את ערכיהם של המשתנים המוגדרים בתכנית במקטע הנתונים. אנו יכולים לרשום את הנתונים בשיטה הבינארית, ההקסדיצמלית או העשרונית ועלינו לציין זאת במפורש. בדוגמה זו בחרנו. להציג את הנתונים בשיטה ההקסדיצמלית ולכן הוספנו ליד כל מספר את האות h.

ההוראה הראשונה שאחריה נעקוב תירשם בשורה השנייה בטבלה ובהמשך תירשם כל הוראה לפי סדר ביצועה.

ההוראה	al	a	b	C
		26h	17h	24h
mov al, a	26h			
add al, b	3Dh			
sub al, c	19h			
mov a, al		19h		



תרגיל 1

ציינו אילו מן ההוראות שלהלן תקינות. אם אינן תקינות, הסבירו מדוע.

- א. mov al, bl
- ב. mov cx, dl
- ג. mov bh, bh
- ד. add 23AFh, bx

```

ה. add cl, 123h
ו. mov dl, 123
ז. mov cx, oDoh
ח. mov 23h, al
ט. sub cl, dl + 3

```

תרגיל 2

מטרת התכנית שלהלן היא להחליף שני נתונים מטיפוס מילה בפולה. הנתון הראשון הוא 12340012h והוא מאוחסן באוגרים BX:AX: המילה העליונה מאוחסנת באוגר BX והמילה התחתונה מאוחסנת באוגר AX. הנתון השני הוא 9876AF01h והוא מאוחסן באוגרים DX:DX: המילה העליונה נמצאת באוגר DX והמילה התחתונה נמצאת באוגר CX. בתכנית חסר חלק מהביטויים. השלימו את הביטויים החסרים, הריצו את התכנית ובדקו אותה.

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

```
.CODE
```

```
start:
```

```

mov ax, @data
mov ds, ax
move ax, -----
move bx, -----
----- dx, 9876h
----- cx, -----
xchg -----, -----
-----
mov ah, 4ch
int 21h

```

```
End start
```

תרגיל 3

התכנית הבאה מבצעת חישוב כלשהו ושמה את התוצאה באוגר DL.

```
.MODEL SMALL
.STACK 100h
.DATA
.CODE
start:

    mov ax, @DATA
    mov ds, ax
    move al, 12
    move bl, 34
    move cl, -30
    move dl, al
    add dl, bl
    add cl, cl
    sub dl, cl
    move ax, 400h
    int 21h
End start
```

- א. עקבו אחרי ביצוע התכנית באמצעות טבלת מעקב ורשמו מהו ערכו הסופי של DL.
- ב. רשמו את הביטוי החשבוני המחושב בטבלת המעקב ואת תוצאת החישוב.

תרגיל 4

- א. כתבו תכנית אשר תשים מספר ללא סימן באוגר CH. התכנית תשתמש בפעולות חיבור בלבד כדי להכפיל את תוכן CH בארבע ותשים את התוצאה באוגר CL.
- ב. חשבו מה הערך הגדול ביותר ללא סימן שאפשר לשים באוגר CH כך שלא תהיה גלישה.
- ג. חשבו מהו הערך החיובי הגדול ביותר עם סימן שאפשר לשים באוגר CH כך שלא תהיה גלישה.

תרגיל 5

כתבו תכנית שתשים במשתנים s ו- c ערכים עוקבים לערך הנמצא ב- a . לדוגמה, אם ערכו של a הוא 7, לאחר ביצוע החישוב ב- s יושם הערך 8 וב- c הערך 9. הגדירו שלושה משתנים מטיפוס בית ואתחלו את המשתנה a לערך 21.

תרגיל 6

רשמו מה תהיה תוצאת החישוב ומה יהיו ערכי הדגלים CF ו- OF לאחר הביצוע של כל אחד מקטעי התכנית שלהלן:

א.

```
mov cl, offh
mov ch, offh
sub cl, ch
```

ב.

```
mov dl, 10000001b
sub dl, 5
```

ג.

```
mov cx, 0a000h
mov bx, 0ffffh
sub cx, bx
```


אוגר הדגלים והוראות קפיצה

חומר רקע

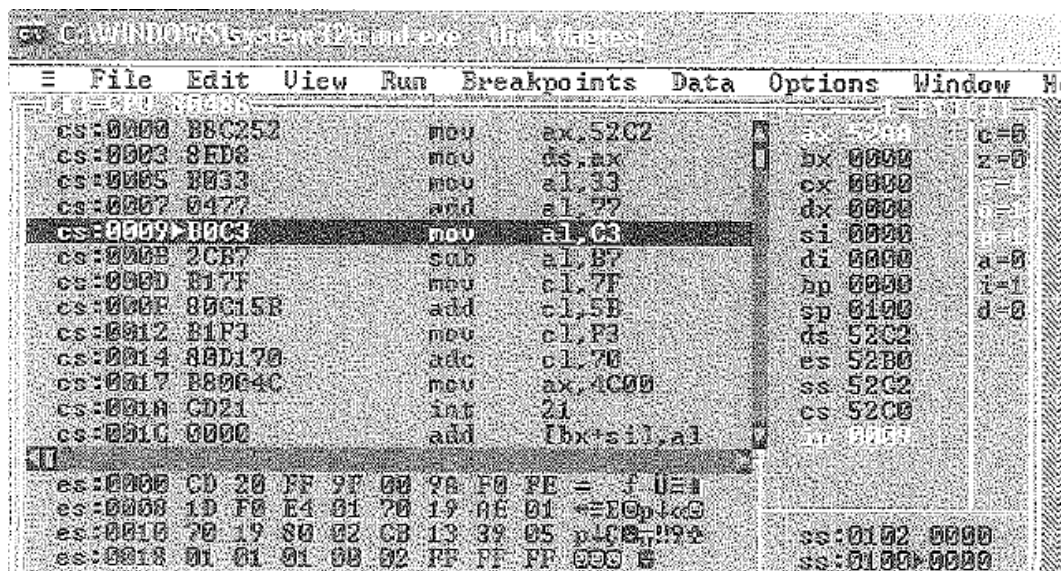
פעילות זו מתייחסת לסעיף 5.3 בספר הלימוד במהלך ביצוע התכנית אפשר לעקוב בעזרת הדיבגר TD אחרי ההשפעה של ביצוע ההוראות השונות על אוגר הדגלים. לדוגמה, נעקוב אחרי מצב הדגלים בביצוע ארבע ההוראות שלהלן:

```
mov al, 033h
add al, 77h
mov al, 0C3h
sub al, 0B7h
```

א. איור 3 מציג את מצב האוגרים לאחר ביצוע שתי ההוראות שלהלן:

```
mov al, 033h
add al, 77h
```

זכרו כי ההוראה MOV אינה משפיעה על מצב הדגלים



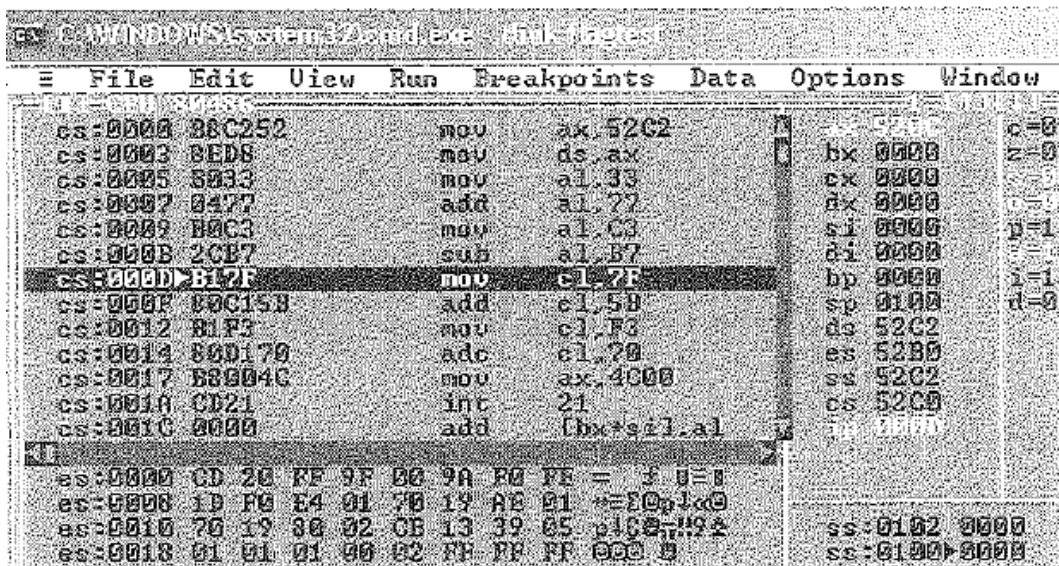
3 איור

מצב הדגלים לאחר ביצוע ההוראה add al, 77h

בתכנית דיבגר TD מצבם של הדגלים שהשתנה בעקבות ביצוע ההוראה מוצג בצורה בולטת.
באיור 3 האוגרים שהושפעו מביצוע ההוראה `add al, 77h` הם: `OF`, `SF` ו- `PF` וערכם נקבע ל- '1'.
ערכם של שאר הדגלים נשאר ללא שינוי והוא '0'.

ב. איור 4 מציג את מצב הדגלים לאחר ביצוע שתי ההוראות הבאות:

```
mov al, 0c3h
sub al, 0b7h
```



4 איור

מצב הדגלים לאחר ביצוע ההוראה `sub al, 0b7h`

באיור 4 אפשר לראות כי ערכם של הדגלים `OF`, `SF` ו- `AF` השתנו כדלהלן: ערכם של `OF` ו- `SF` נקבע ל-'1' ואילו ערכו של `AF` הוא '0'.



תרגיל 1

א. התבוננו באיורים 3 ו-4. האם אפשר לדעת מה יהיה ערכם של הדגלים לאחר ביצוע ההוראה:

```
mov al, 0C3h
```

אם כן, רשמו את ערכם של דגלי המצב לאחר ביצוע ההוראה זו.

ב. רשמו מה יהיה ערכם של דגלי המצב לאחר ביצוע ההוראות הבאות:

```
mov cl, 7Fh  
add cl, 5Bh  
mov cl, 0F3h  
adc cl, 70h
```

תרגיל 2

התבוננו בתכנית הבאה:

```
.MODEL SMALL  
.STACK 100h  
.DATA  
.CODE  
start:
```

```
mov ax, @DATA  
mov ds, ax  
mov ah, 0Ah  
mov bl, 4  
mov bh, 5  
mov cl, 5  
mov dl, 0
```

```

add, bl, bl
cmp bl, ah
jnz cont1
inc dl
cont1: add bh, bh
cmp bh, ah
jnz, cont 2
inc dl
cont2: add cl, cl
cmp cl, hl
jnz sof
inc dl
sof: mov ah, 4ch
int 21h

```

END start

- א. עקבו באמצעות טבלת מעקב אחדי ביצוע התכנית ורשמו מהו ערכו הסופי של האוגר DL.
- ב. טבלת המעקב הקצו עמודה לכל אוגר שבו אתם משתמשים בתכנית ועמודה נוספת לתיאור מצב דגל האפס ZF לאחר ביצוע הוראות cmp.
- ג. מהי מטרת התכנית?
- ד. נסחו במילים את התנאי הלוגי שממומש בהוראות:

```

cmp bl, ah
jnz cont1

```

תרגיל 3

תלמיד כתב תכנית הבדקת אם המספר הנמצא באוגר CL שונה משמונה ואם כן, היא שמה את הערך 'ו' באוגר DL. אם ערכו של CL הוא שמונה היא שמה באוגר DL את הערך 'ס';

להלן התכנית:

.MODEL SMALL

.STACK100h

.DATA

.CODE

start:

אתחול מקטע נתונים:

```
mov ax, @data
mov ds, ax
mov cl, 12
cmp cl, 8
jnz NotEqual
mov dl, 0
NotEqual : mov dl, 1
```

סיום התכנית:

```
mov ah, 4ch
int 21h
```

End start

- התכנית שכתב התלמיד שגויה והיא אינה משיגה את מטרתה.
- האם עבור הערך הרשום באוגר CL התכנית מתבצעת כראוי?
 - רשמו מספר שעבורו התכנית מתבצעת בצורה שגויה.
 - תקנו את התכנית.

תרגיל 4

כתבו תכנית הבודקת אם המספר באוגר AX מכיל את הסיבית 1 לפחות פעם אחת. אם כן, שימו באוגר DL את הערך '5' ואם לא – את הערך '4'.

תרגיל 5

להלן קטע של תכנית הבודקת אם הערכים שבאוגרים D1 ו D4 הם תקינים.
הגדרה: באוגר ערך הוא תקין אם הוא מציל את הסיבית 1 לפחות פעם אחת.
התכנית בודקת אם שני האוגרים "תקינים". ואם כן – היא שמה את הערך '1' באוגר A1.
אם לא – היא שמה את הערך '0' באוגר A1.

Start:

```
mov al, 0 ; הנחה - האוגרים אינם "תקינים";  
mov dl, 0 ; dl=0  
mov dh, 5eh ; dh = 5Eh  
cmp dx, 0 ; האם DX מכיל לפחות סיבית שאינה 0 ?
```

```
jnz sof  
mov al, 1
```

sof:

סיום התכנית ;

- הקטע המוצע אינו נכון.
א. כתבו דוגמה נוספת שלפיה הפתרון נכון.
ב. כתבו דוגמה המראה כי הפתרון אינו נכון.
ג. תקנו את התכנית.
ד. האם אפשר למחוק את ההוראה הראשונה: `mov al, 0`?

תרגיל 6

להלן תכנית המבצעת בדיקה כלשהי ושמה את התוצאה במשתנה `check`.

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

```
char1 db 'z'  
char2 db 'e'  
check db 0
```

```
.CODE
```

start:

```
mov ax, @data  
mov ds, ax  
mov cl, 'e'  
cmp cl, char1  
jnz checkChar  
jmp sof
```



```

checkChar:    cmp cl, char2
              jnz sof
              mov check, 1
sof:          mov ah, 4ch
              int 21h
              END Start

```

- א. השתמשו בטבלת מעקב ורשמו מהי מטרת התכנית.
 ב. נסחו במילים תנאי המתאים לבדיקה המתבצעת בהוראות הבאות:

```

cmp cl, char2
jnz sof

```

תרגיל 7

ציינו אילו מן ההוראות שלהלן תקינות תחבירית ואילו אינן תקינות. הניחו כי כל המשתנים הם מטיפוס בית.

- א. `mov al, num`
 ב. `mov cl, num + num1`
 ג. `mov num, num1`
 ד. `cmp bl, 12`
 ה. `cmp num, 0`
 ו. `cmp ax, dx`
 ז. `cmp num1, num`
 ח. `cmp bl, num + 3`
 ט. `add ch, num`
 י. `add num1, 4`
 יא. `add num1, num`
 יב. `mov ip, 10h`
 יג. `mov ir, 12h`
 יד. `mov ir, cl`

תרגיל 8

ידוע כי במשתנה char1 מטיפוס בית יש אות אנגלית גדולה.
כתבו תכנית השמה במשתנה char2 את האות העוקבת לאות שבמשתנה char1.
לדוגמה: אם נתון כי 'Char1='C', אזי ב - char2 יושם 'D'.
בתכנית הרשומה למטה חסרים מספר ביטויים. השלימו אותם.

שימו לב! האות העוקבת לאות 'Z' היא האות 'A'

```
.MODEL SMALL
.STACK 100h
.DATA
    char1 db 'C'
    char2 db ?
.CODE
start:
    mov ax, @data
    mov ds, ax
    cmp char1, -----
    je xx
        mov char2, char1
        -----
        jmp finish
xx:      -----
finish:
    mov ah, 4ch
    int 21h
END start
```

תרגיל 9

מה תהיה תוצאת החישוב ומה יהיו ערכי הדגלים CF ו - OF לאחר הביצוע של כל אחד מקטעי התכנית שלפניכם:

א.

```
mov clo, ffh
mov cho, fh
sub cl, ch
```

ב.

```
mov dl, 10000001b
sub dl, 5
```

ג.

```
mov cx, 0a000h
mov bx, 0ffffh
sub cx, bx
```

תרגיל 10

רשמו מה יהיה מצב הדגלים ZF ו CF – לאחר ביצוע כל אחד מקטעי התכנית שלהלן:

א.

```
mov ax, 30h
cmp ax, 20h
```

ב.

```
mov dl, 10010
mov cl, 12h
cmp dl, cl
```

ג.

```
mov bx, 100h
add bx, 1000000000b
```

תרגיל 11

כתבו תכנית המונה כמה מהמספרים הנמצאים במשתנים N_1 , N_2 , N_3 הם מספרים שליליים. את המונה שימו במשתנה MoneNeg.

תרגיל 12

במשתנה AsciiChar נתון תו.
התכנית הרשומה למטה פותרת את הבעיה הבאה :

אם AsciiChar מכיל ספרה(כלומר, תו בין '0' ל-'9')
אזי נשים במשתנה Digit את הספרה הזו
אם לא - נשים במשתנה Digit את התו '?'

השלימו את המקומות הריקים.

SMALL MODEL.

100h STACK.

DATA.

asciiChar DB '7'

digit DB '?'

CODE.

mov ax, @data

mov ds, ax

mov dl, AsciiChar

cmp dl, '0'

cmp '9', dl

Js sof

sub dl, '0'

sof: mov ah, 4ch

int 21h

END Start

תרגיל 13

כתבו תכנית שתשים במשתנים char1 ו char2 שני תווים שהם אותיות גדולות באנגלית. התכנית תשים במשתנה mone את כל התווים שישנם בין char1 ו char2 (לרבות char2).

לדוגמה:

אם $\text{char1} = 'A'$ $\text{char2} = 'D'$ נקבל $\text{mone} = 4$
ואם $\text{char1} = 'F'$ $\text{char2} = 'B'$ נקבל $\text{mone} = 5$

תרגיל 14

נתון כי טמפרטורת ההתכה של ברזל היא 1535 מעלות צלסיוס, טמפרטורת ההתכה של עופרת היא 328 מעלות צלסיוס וטמפרטורת ההתכה של כסף היא 962 מעלות צלסיוס. כתבו תכנית המוצאת את טמפרטורת ההתכה הגבוהה ביותר של שלוש המתכות הנתונות.

א. הגדירו ארבעה משתנים: בשלושה מהם יאוחסנו נתונים על טמפרטורת ההיתוך של שלושת המתכות ובמשתנה הרביעי תאוחסן הטמפרטורה הגבוהה ביותר.

ב. כתבו תכנית מתאימה.

ג. האם התכנית שכתבתם בסעיף ב' תהיה תקינה אם במקום בטמפרטורת ההתכה של המתכת כסף, נרשום את טמפרטורת ההתכה של היסוד חמצן שהיא 219- מעלות צלסיוס ושאר הנתונים יישארו ללא שינוי? אם לא, בצעו את השינויים הדרושים וכתבו תכנית מתאימה.

תרגיל 15

נניח כי אוגרים AX ו BX מכילים מספרים עם סימן ואוגרים CX ו DX מכילים מספרים ללא סימן. רשמו אילו תנאים יש לרשום לאחר הוראת CMP כדי לבדוק האם:

1. תוכנו של DX גדול מתוכנו של CX .
2. תוכנו של BX גדול מתוכנו של AX .
3. תוכנו של DX הוא 0.
4. תוכנו של BX גדול או שווה לתוכנו של AX .
5. תוכנו של CX קטן או שווה לתוכנו של DX .

תרגיל 16

כתבו קטע תכנית למימוש התנאי הבא:

האם $\text{AX} > 3$ וגם $\text{BX} < 2$?

תרגיל 17

כתבו תכנית הבודקת את מספר התכונות המתקיימות במספר עם סימן המאוחסן במשתנה `var`. התכונות הנבדקות הן אלה:

1. אם המספר חיובי.
2. אם המספר הוא 0.
3. אם המספר הוא זוגי.
4. אם כל הסיביות שבו הן 1.

- א. כתבו תכנית המאחסנת במשתנה `perfect` את הערך '1' אם המספר מקיים את כל התכונות הנבדקות.
- ב. כתבו תכנית המאחסנת במשתנה `result` את מספר התכונות המתקיימות במספר הנבדק.

לולאות

חומר רקע

פעילות זו מתייחסת לסעיף 5.4 בספר הלימוד

נדגים שימוש בטבלת מעקב לקטע התכנית שלהלן, המחשב את הפעולה $X \cdot Y$ ומאחסן את התוצאה במשתנה prod מטיפוס בית (בהנחה שתוצאת המכפלה אינה גולשת). נתון כי X ו- Y הם מספרים לא מסומנים מטיפוס בית.

$$X \cdot Y = X + X + X + \dots + X$$

את הסכום מבצעים Y פעמים.

	mov cx, 0	; אתחול מונה לולאה cx
	mov cl, y	; cx = y
cont:	mov dl, 0	; DL יכיל את הסכום
again:	add dl, x	; DL יכיל את $x \cdot y$
	loop again	
	mov prod, dl	

להלן טבלת המעקב של קטע התכנית עבור הערכים $x = 3$ ו- $y = 4$. שימו לב! ההוראה loop מפחיתה '1' מערכו של האוגר CX המוצג כשני אוגרים מטיפוס בית: האוגר CL והאוגר CH.

ההוראה	X	Y	CH	CL	DL	prod
אתחול משתנים	3	4				
mov exo ,			0	0		
mov cl ,y				4		
mov dlo ,					0	
add dl ,x					3	
loop again				3		
add dl ,x					6	
loop again				2		
add dl ,x					9	
loop again				1		
add dl ,x					12	
loop again				0		
mov prod ,dl						12



תרגיל 1

- קטע התכנית לחישוב $y \cdot x$ אינו מטפל במקרה אחד. מהו המקרה?
- א. תקנו את קטע התכנית כך, שהוא יטפל בכל המספרים הלא מסומנים מטיפוס בית שתוצאת המכפלה שלהם תהיה גם היא מטיפוס בית.
- ב. אם ערכו של x הוא עשר, מהו ערכו המכסימלי של y כך, שהמכפלה $y \cdot x$ תהיה עדיין בגודל בית.
- ג. שנו את התכנית כך, שהיא תחשב נכונה את הכפל של כל שני מספרים x ו- y שהם מטיפוס בית. שימו לב! במקרה זה התוצאה יכולה להיות מטיפוס מילה.

ד. כדי לייעל את התכנית כדאי לשים במונה CX את הערך הקטן מבין X ו- Y ולחבר את הערך הגדול מביניהם. לדוגמה, אם נתון כי $Y = 5$ ו- $X = 12$, כדאי לבצע את החישוב:

$$12+12+12+12+12 = 5 \cdot 12$$

שנו את התכנית שכתבתם בסעיף ד' כך, שתבצע את החישוב ביעילות.

תרגיל 2

א. כתבו תכנית המחשבת את סכום עשרת האיברים הראשונים בסדרת פיבונאצ'.

הגדרה של שדרת פיבונאצ' – שני האיברים הראשונים בסדרה הם 0 ו-1 וכל איבר החל מהאיבר השלישי הוא סכום של שני האיברים הקודמים לו.

ב. שנו את התכנית שכתבתם בסעיף א' כך, שהיא תחשב את סכום האיברים הראשונים בסדרת פיבונאצ' כל עוד הסכום קטן מ-100. בסיום נאחסן במשתנה `sum` את הסכום שחושב ובמשתנה `count` את מספר האיברים שסוכמו.

תרגיל 3

כתבו תכנית הבודקת אם משתנה מטיפוס מילה הנו מספר ראשוני. אם המספר הוא ראשוני, יאוחסן הערך '1' באוגר `BX`. אחרת, יושם בו הערך '0'.
רמז: השתמשו בפעולת חיסור לביצוע פעולת חילוק.

תרגיל 4

להלן קטע תכנית שאמור לממש את האלגוריתם הבא:

אם ערכו המוחלט של `al` הוא כפולה של 10
אזי נשים באוגר `ah` את המספר 1
אחרת, נשים באוגר `ah` את המספר 0.

השלימו את המקומות הריקים.

```

-----
mov al, 40
cmp al, 0
je finish

-----

neg al
cont:
jnl cont

-----

dec ah
mov ah, 4ch
finish:
int 21h

```

תרגיל 5

כתבו תכנית בשפת סף עבור האלגוריתם הבא:

$X = 0$
 2. עבור i מ-1 עד num בצע
 2.1 עבור j מ-1 עד $i + 1$ בצע
 2.1.1 $X = X + 1$

הניחו כי ערכו של num קטן מ-25.

תרגיל 6

כתבו תכנית המגדירה ארבעה משתנים מטיפוס בית: n , x , x_1 ו- $diff$. התכנית מחשבת את ערכו של האיבר i -י בסדרה חשבונית שבה האיבר הראשון הוא x_1 וההפרש שבין שני איברים עוקבים הוא $diff$. התוצאה תאוחסן במשתנה x . הניחו כי משתנה n הוא מספר ללא סימן השונה מ-0, ואילו x_1 ו- $diff$ הם מספרים עם סימן. כמר כן אפשר להניח כי ערכו המחושב של x הוא גם כן מטיפוס בית.

תרגיל 7

נשנה את ההנחה שערכו המחושב של x במשימה 6 הוא מטיפוס בית ותיתכן גלישה. שנו את התכנית שכתבתם כך, שבסיומה x יכיל את הערך האחרון בסדרה שהוא בגודל בית.

התכנית תאחסן במשתנה ח את המיקום של האיבר חא בסדרה.

תרגיל 8

א. כתבו טבלת מעקב אחר ביצוע קטע התכנית הבא:

```
mov al, 0
mov cx, 3
mov bl, 1
label1: add al, bl
        inc bl
        loop label1
```

בטבלת המעקב רשמו עמודות מתאימות לכל אחד מהאוגרים המופיעים בקטע התכנית.

ב. רשמו מהו ערכו הסופי של אוגר la

ג. רשמו מה מבצעת התכנית.

תרגיל 9

הוסיפו הוראות לקטע התכנית שהוצג בתרגיל 8 כך, שביצוע הלולאה יתבצע ארבע פעמים.

האוגר dx מכיל את מספר הפעמים שקטע התכנית בתרגיל 8 יתבצע.

צריך להשלים הוראות ולמקם את התווית label2

```
mov al, 0
mov dx, 4
mov cx, 3
mov bl, 1
label1 : add al, bl
        inc bl
        loop label1
```

```
-----
-----
jmp label2
```


הוראות לוגיות

חומר רקע

פעילות זו מתייחסת לסעיף 5.5 בספר הלימוד



תרגיל 1

כתבו טבלת מעקב אחר ביצוע הקטעים הבאים:

א.

```
mov ax, 0001110001110001B
mov bx, 1110001110001111B
and ax, bx
not ax
or ax, bx
not bx
jz newtry
and bx, ax
not bx
jmp exit
newtry : and ax, bx
exit :not ax
```

```

        .b
        mov al, 23h

        mov ch, 68h
        xor al, ch
        test al, 77h
        jz missand
        and ch, 33h
missand: xor ch, 0AAh
        test ch, al
        jz misor
        not al
        or ch, al
misor: not ch

```

תרגיל 2

בספרייה עורכים סקר על דירוג עשרת הספרים הממוספרים מ-1 עד 10.
כל קורא מסמן בטופס את מספרי הספרים שהוא קרא.
יש לכתוב תכנית הבדקת אם:

א. הקורא קרא את ספר מספר חמש.

ב. הקורא קרא ספר אחד בלבד.

ג. הקורא קרא בין שניים לחמישה ספרים.

ד. הקורא קרא חמישה ספרים ויותר.

שמרו את תוצאות הבדיקה (ב"כן" או "לא") במשתנים.

הגדירו תחילה את המשתנים והאוגרים שבהם תשתמשו, את הטיפוס שלהם ואת תפקידם ואחר כתבו אלגוריתם מילולי ותכנית מתועדת.

תרגיל 3

בחברה להשכרת דירות רושמים לגבי כל דירה, את המידע הזה:

- האם יש בדירה דוד שמש
- האם יש בדירה מיזוג אוויר
- האם זהו בית פרטי או שהדירה נמצאת בבית דירות
- האם יש בדירה ריהוט
- האם יש במטבח הדירה מכשירי חשמל
- האם יש חנייה
- האם יש גינה

כל לקוח מסמן בטופס אילו מן הפרטים האלה הכרחיים בדירה שהוא מבקש לשכור.
כתבו תכנית הבדקת:

- א. האם הדירה מתאימה לדרישות לקוח.
- ב. האם הדירה מתאימה ללקוח המוכן להתפשר על הפרטים להלן: ריהוט ומכשירי חשמל (שיתכן שיהיו ויתכן שלא יהיו בדירה) אך דורש ששאר הפרטים יתקיימו?
- ג. אם הדירה אינה מתאימה לדרישות הלקוח, מנו כמה פרטים בדירה מסוימת מתאימים לדרישות הלקוח?

בפתרון, הגדירו את שמות המשתנים והאוגרים שבהם תשתמשו, את הטיפוס שלהם ואת תפקידם.

כתבו אלגוריתם מילולי ותכנית מתועדת.

הוראות הזזה והוראות סיבוב

חומר רקע

פעילות זו מתייחסת לסעיף 5.7 בספר הלימוד

נכתוב טבלת מעקב לקטע התכנית המונה את מספר הסיביות שערכן '1' במספר הנמצא באוגר al.
מספר הסיביות מושם באוגר ah.

xor ah, ah	; אפס את המונה ah
mov al, 25h	; שים h25 באוגר al
again: cmp al, 0	; האם 0 = al
je finish	; al = 0 ולכן סיים התכנית
shr al, 1	0 < al ולכן הזז שמאלה סיבית אחת
jnc again	אם הסיבית השמאלית היא 0 חזור
inc ah	ולכן הגדל מונה 0 הסיבית השמאלית
jmp again	חזור על הפעול ה
finish :	סיום

להלן טבלת המעקב עבור התכנית.

ההוראה	AL	AH	CF	התנאי הנבדק
xor ah ,ah		0		
mov al5 ,h	5h = 00000101			
cmp alo ,	00000101			האם 0 = al?
je finish				0 = al ולכן התנאי נכשל
shr al1 ,	00000010		1	
jnc again				הסיבית שנפלה 0 ולכן המשך להוראה הבאה
inc ah		1		
jmp again				
cmp alo ,				האם 0 = al?
je finish				al = 0 ולכן המשך להוראה הבאה
shr al1 ,	00000001		0	
jnc again				הסיבית שנפלה 0 = ולכן עבור again
cmp alo ,				האם 0 = al?
je finish				al = 0 ולכן המשך להוראה הבאה
shr al1 ,	00000000		1	
jnc again				הסיבית שנפלה 0 ולכן המשך להוראה הבאה
inc ah			2	
jmp again				
cmp alo ,				האם 0 = al?
je finish				al = 0 ולכן עבור ל - finish



תרגיל 1

שנו את התכנית המופיעה לעיל כך, שהיא תחשב את מספר הסיביות באוגר AL שהן 0.

תרגיל 2

ענו על השאלות הבאות:

- א. האם ההוראות `shl dl, 4` ו- `rol dl, 4` ייתנו אותה תשובה? אם כן, רשמו את המספר באוגר `dl`.
- ב. האם ההוראות `shr cl, 2` ו- `ror cl, 2` ייתנו אותה תשובה? אם כן, רשמו את המספר באוגר `cl`.
- ג. האם ההוראות `shl dh, 4` ו- `rcl dh, 4` ייתנו אותה תשובה? אם כן, רשמו את המספר באוגר `dh`.
- האם ההוראות `rol al, 3` ו- `rcl al, 3` ייתנו אותה תשובה? אם כן, רשמו את המספר באוגר `al`.

תרגיל 3

באוגר `DX` יש מספר כלשהו. כתבו הוראות המחליפות בין הבית התחתון והבית העליון של האוגר. לדוגמה: אם ערכו התחילי של `DX` הוא `5A9Ch`, לאחר ההחלפה ערכו יהיה `9c5Ah`.

באוגר `AL` יש מספר כלשהו. כתבו הוראות המחליפות את ארבע הסיביות התחתונות של האוגר עם ארבע הסיביות העליונות שלו. לדוגמה: אם ערכו התחילי של `al` הוא `82h`, לאחר ההחלפה ערכו יהיה `28h`.

תרגיל 4

קטע התכנית הרשום להלן מוצא את ההופעה הראשונה (החל מצד ימין) של הסיבית 0 במספר המאוחסן במשתנה num המוגדר מטיפוס מילה. את מיקום הסיבית יש לשמור במשתנה בשם index. האם התכנית נכונה? אם לא, תקנו אותה.

```

mov index, 0
mov ax, num
cmp ax, 0
jz sof
again : inc index
      shr ax, 1
      jc again
finish:

```

תרגיל 5

שימו באוגרים ah, al, bl, bh מספרים. כתבו תכנית הבודקת אם בכל האוגרים מופיעה סיבית 0 בחצי השמאלי של המספר לפחות פעם אחת. אם כן, שימו במשתנה check את הערך '10'. אחרת, שימו במשתנה check את הערך '10'.

תרגיל 6

כתבו תכנית המוצאת כמה פעמים סיבית 1 מופיעה במקומות זוגיים במספר Num שגודלו מילה. את התוצאה שימו במשתנה mone.

תרגיל 7

באוגר al יש מספר. כתבו תכנית המכפילה את תוכן al ב-14 בתנאי שהמספר הנו חיובי וזוגי.

תרגיל 8

להלן תכנית המוצאת כמה פעמים הסיבית 0 מופיעה באוגרים al ו ah באותם מקומות.
התוצאה מאוחסנת באוגר dl.
דוגמה : al = b01001011
ah = b11011010
התוצאה dl = 2

```
-----  
mov al,01100101 ,b  
mov ah, 01010011 b  
mov cx, 8
```

again:

```
-----  
shr al, 1  
adc bl, 0
```

```
-----  
jne cont  
inc dl
```

cont:

```
-----  
mov ah,4 ,ch  
int21 h
```

תרגיל 9

כתבו תכנית השמה באוגר dl מספר. התכנית בודקת אם החצי השמאלי של המספר הוא שיקוף של החצי הימני. אם כן, שימו באוגר dh את הערך '1'. אחרת, שימו באוגר dh את הערך '0'.
דוגמה dl = 10111010 לאחר הרצת התכנית 1 = dh.

תרגיל 10

כתבו תכנית למערכת הבאה:

במערכת בקרה במפעל קיימים שמונה חיישנים המודדים את מצבן של שתי מכונות. ארבעה חיישנים בודקים את מצבה של מכונה א', וארבעה חיישנים בודקים את מצבה של מכונה ב'.

מערכת הבקרה מציגה את מצב המכונות באמצעות המשתנים $checka$ ו $checkb$. אם אחד החיישנים המחוברים למכונה א' מוציא את הערך '1' יש לשים במשתנה $checka$ את הערך '1'. אחרת, יש לשים במשתנה את הערך '0'. אותה פעולה יש לבצע גם לגבי מכונה ב' אך את התוצאה יש לשים במשתנה $checkb$.

תרגיל 11

כתבו תכנית השמה באוגר bl מספר מסומן. התכנית מחלקת את המספר בשתיים עד שמתקבל מספר אי-זוגי. את התוצאה נשים באוגר bh . אם ידוע כי אף פעם לא נקבל מספר אי-זוגי אזי העתיקו ל bh את המספר הנמצא ב bl .

דוגמה : $bl = b01110100$ התוצאה : $bh = b00011101$

12 (רשות) תרגיל

"במקביל"

עד כה, תיארונו שיטה סטנדרטית לחיבור מספרים. ואם נרצה לממש אותה במחשב, הרי שנשתמש במעגל אלקטרוני שיהווה "יחידת חיבור", אשר במחשב תהיה יחידה בודדת כזו לחיבור של סיבית בודדת ונשא. ביחידה זו נשתמש בכל פעם לחיבור הספרה הבאה בתור והנשא שהתקבל מהחיבור הקודם. אבל במערכות חישוב מתקדמות מקובל לבצע חיבור של שני מספרים בדרך אחרת, שהיא בדרך כלל מהירה יותר. נתאר תחילה את השיטה כאילו אנחנו מחשבים בכתב, ואחר נציגה באופן כללי כפי שהיא ממומשת במחשב.

אם נשתמש בשיטה זו בכתב, נבצע אותה על פי האלגוריתם הבא:

1. מחברים כל אחת מן הספרות של המספר הראשון עם הספרה המתאימה לה במספר השני. אם יש נשא, כותבים אותו בשורה שמתחת לשורת התוצאה, כשהוא מוסט במקום אחד שמאלה.
2. אם קיבלנו שורת נשא, משתמשים בתוצאה ובשורת הנשא כשני המספרים החדשים לחיבור, וחוזרים לשלב 1 ; אחרת - סיימנו.

כך חוזרים על שני ש' לביס אלה עד ששורת הנשא נותרת ריקה. נמחיש את התהליך באמצעות דוגמה שבה נחבר שני מספרים בני 6 סיביות:

$$\begin{array}{r}
 +010001 \\
 100011 \\
 \hline
 \begin{array}{r}
 +110010 \quad \text{הסכום} \\
 1 \quad \text{הנשא} \\
 \hline
 +110000 \quad \text{הסכום} \\
 1 \quad \text{הנשא} \\
 \hline
 110100
 \end{array}
 \end{array}
 \begin{array}{l}
 \text{שלב ביניים I :} \\
 \text{שלב ביניים II :} \\
 \text{התוצאה:}
 \end{array}$$

בשלב הראשון, התקבל נשא מחיבור שתי הספרות שנמצאות במיקום השני מימין. תוצאת החיבור שלהן היא $10_2 = 1+1$. את הסיבית 0 אנו רושמים באותו טור מתחת לספרות שחיברנו, ואת הסיבית 1 שהיא הנשא, אנו רושמים בטור הסמוך משמאל.

כיוון שבחיבור אחד הזוגות התקבל נשא, אנו חוזרים ומבצעים פעם נוספת את שלב 1. הפעם לא התקבל שום נשא, ולכן התוצאה שהתקבלה, היא התוצאה הסופית.

היתרון של שיטה זו על השיטה הסטנדרטית, הוא שהמימוש שלה מאפשר לטפל במקביל בכל זוגות הסיביות וכך לקצר את זמן הפעולה. כלומר, לכל זוג סיביות מוקצית יחידת חיבור: היחידה הראשונה מחברת את הסיביות במיקום 0; באותו זמן יחידה שנייה מחברת את הסיביות שמיקומן 1, יחידה שלישית מחברת את הסיביות שמיקומן 2 וכך הלאה. בצורה זו, בכל שלב, אנו יכולים להפעיל, באותו זמן (במקביל) את כל שש יחידות החיבור וכך לחבר את כל הסיביות של שני המספרים, בזמן הדרוש לחיבור זוג אחד של סיביות. כך קיצרנו את ביצוע החיבור של שני המספרים שבדוגמה בשני שלבים. לעומת זאת, בשיטה הקודמת טיפלנו בסיביות המספר בצורה סדרתית, ובכל שלב חיברנו רק 2 סיביות (ונשא אם היה). לכן ביצוע החיבור של שני המספרים הללו ארך שישה שלבים. כלומר, בדוגמה שלנו, זמן החיבור בשיטה הסטנדרטית ארוך פי 3 מאשר זמן החיבור של אותם מספרים בשיטה המקבילית.

א. האם השיטה "המקבילית" לחיבור שני מספרים היא תמיד יותר מהירה מאשר השיטה הסטנדרטית? אם אתם חושבים שלא כך הדבר, תנו דוגמה למקרה שבו אלן יתרון לשיטה זו, וציינו באילו מקרים יש לה יתרון.

ב. כתבו תכנית המממשת את האלגוריתם לחיבור שני מספרים בינאריים
"במקביל". בשיטה זו, יש שתי פעולות בסיסיות:

- בפעולה הראשונה מחברים כל אחת מן הספרות של המספר הראשון עם הספרה המתאימה לה במספר השני. אם יש נשא, כותבים אותו בשורה שמתחת לשורת התוצאה, מוסט במקום אחד שמאלה.
- בפעולה השנייה מחברים את שורת הנשא לשורת התוצאה. כך חוזרים על שתי פעולות אלה עד ששורת הנשא נותרת ריקה.

רמז: בכל שלב השתמשו בפעולה לוגית XOR לחיבור הספרות ובפעולה הלוגית AND לחישוב הנשא.

תרגיל 13

מכונה לממבר משקאות מחזירה עודף ב- 10 אג' וב- 50 אג'.
המכונה מקבלת את הסכום לתשלום ואת מחיר משקה, מחשבת עודף ואת מספר המטבעות המינימלי שיש להחזיר מכל סוג.

לדוגמה, אם מחיר המשקה הוא 6.80 ש' ולמכונה הוכנסו 10 ש', העודף שעליה להחזיר הוא 3.20 ש':
שש מטבעות של 50 א"ג ושתל מטבעות של 10 א"ג.

רמז: ייצגו את הסכום כאגורות. לדוגמה, במקום לרשום 3.20 ש' רשמו 320 א"ג.

שיטות מיעון ומערכים

חומר רקע

פעילות זו מתייחסת לפרק 6 בספר הלימוד



תרגיל 1

כתבו תכנית המגדירה מערך בן עשרה איברים מטיפוס בית ואשר מאתחלת אותו

לערכים הבאים:

א. `oFh, 45h, oA1h, 78h, oCFh, oAah, 8Fh, 19h`. התכנית מחשבת ומוצאת:

ב. את מספר המספרים הזוגיים במערך.

את המספר הגדול ביותר ואת מיקומו במערך.

התכנית מאחסנת את התוצאות במשתני עזר.

תרגיל 2

כתבו תכנית הבודקת כי איברי כל המערך מטיפוס בית שעליו מצביע המשתנה `first` מקיימים את התנאי הבא: הסיבית השלישית מימין היא 1 וגם הסיבית השישית היא 0. מספר איברי המערך מאוחסן במשתנה `count`. אס הבדיקה הסתיימה בהצלחה שימו במשתנה `result` את המספר 5. אם לא, שימו את המספר -5.

תרגיל 3

לכל הוראה שרשמתם בתכנית שבתרגיל 2 רשמו את שיטת המיעון שהוראה זו משתמשת בה.

תרגיל 4

הקטע הרשום למטה אמור לפתור את הבעיה הבאה:
יש להעתיק את תוכן המשתנה Place מטיפוס בית למערך Buffer אשר בו עשרה איברים מטיפוס בית.
הפתרון אינו נכון. רשמו תחילה מהי השגיאה ואחר כך רשמו הוראות מתאימות כך, שהפתרון יהיה נכון.

```
mov al, byte ptr Place
lea si, buffer
mov cx, 10
again: mov [si + cx], al
loop again
mov ah, 4ch
```

תרגיל 5

נתונה סדרה של ציונים שבה כל ציון הוא מספר בין 0 ל-10. על התכנית לחשב את מספר המספרים מכל ציון. הציונים נמצאים במערך שעליו מצביע המשתנה marks.
הציון האחרון מסומן כ-1-.

תרגיל 6

רשמו מה מטרת קטע התכנית הרשום מטה:

```
mov bx, 10h
mov dl, 08h
mov al, [bx]
and al, 0f0h
jz sof
again: shl dl, 1
test al, dl
jz again
finish: shr dl, 4
mov ds: [20h], dl
```

תרגיל 7

התכנית הבאה מוסיפה ערך קבוע לכל איבר במערך חד-ממדי מטיפוס מילה.
א. הוסיפו הוראות המתאימות כך, שהגישה לאיברי המערך תהיה בשיטת מיעון:
ב. אוגר עקיף
ג. בסיס
אינדקס-בסיס

```
.MODEL SMALL  
.STACH 100h
```

```
    N EQU 7  
    NUMBER EQU 12
```

```
.DATA  
    array DW 62, 74, 50, 70, 80, 80, 90
```

```
.CODE
```

```
start:
```

```
    mov ax, @DATA  
    mov ds, ax  
    mov cx, N
```

```
again: add _____, NUMBER
```

```
        _____
```

```
    Loop again
```

```
finish: mov ax, 4C00h
```

```
    int 21h
```

```
END start
```

תרגיל 8

התכנית הבאה בודקת האם המערך שעליו מצביע first ממוין בסדר עולה. המערך הוא מטיפוס בית ומכיל מספרים עם סימן. האיבר האחרון בו הוא המספר 0. אם המערך ממוין, התכנית שמה את הערך '1' במשתנה Check. אחרת, ערכו הוא '0'. בדקו אם התכנית נכונה.

א. אם התכנית אינה נכונה הסבירו מדוע ותקנו את השגיאה.

ב. רשמו באיזו שיטת מיעון יש לפנות לאיבר במערך.

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

```
first DB 3h, 17h, 15h, 34h, 0
```

```
check DB 0
```

```
.CODE
```

```
start:
```

```
mov ax, @DATA
```

```
mov ds, ax
```

```
lea bx, first
```

```
mov si, 0
```

```
again: mov al, [bx+si]
```

```
cmp al, 0
```

```
je ok
```

```
cmp al, [bx+si+1]
```

```
jge sof
```

```
inc bx
```

```
inc si
```

```
jmp again
```

```
ok: inc check
```

```
sof: mov ax, 4C00h
```

```
int 21h
```

```
END start
```

תרגיל 9

נתון מערך Numbers המכיל מספרים בגודל בית. ידוע כי המספר האחרון במערך הוא 00. כתבו תכנית המחליפה את האיבר הראשון במערך עם המספר הראשון הגדול ממנו והנמצא באיברים העוקבים לו. אם אין במערך מספר אחר הגדול מהאיבר הראשון ערכו לא ישתנה. דוגמה: המערך Numbers לפני ביצוע התכנית

0	1	2	3	4
18h	6h	22h	0Ah	0h

בסוף התכנית ייראה מערך Numbers כך:

0	1	2	3	4
22h	66h	18h	0Ah	0h

תרגיל 10

כתבו תכנית שתבדוק אם מערך מטיפוס בית מתאר רשימה מקושרת מעגלית. כל איבר במערך יכול מספר שיהיה אינדקס של איבר אחר במערך. האיבר האחרון יכול תמיד את האינדקס 0 (של האיבר הראשון). לדוגמה:

0	1	2	3	4	5	מס.איבר
4	3	1	5	2	0	היסט

אם המערך יתאר רשימה מעגלית, שימו את הערך '1' במשתנה ok. אם לא, שימו בו את הערך '0'. הגדירו משתנה FirstAddress שיצביע על האיבר הראשון במערך, משתנה length שיכיל את מספר איברי המערך ומשתנה LastAddress שיצביע על האיבר האחרון בו.

תרגיל 11

מפעל מוביל את מוצריו ללקוחותיו באמצעות רכב המפעל שבו מותקן מד מרחק בן ארבע ספרות הקסדצימליות. במפעל מחשבים את מספר הקילומטרים שהרכב נסע במשך חודש ולשם כך משתמשים בתכנית בשפת אסמבלי. אחרי כל נסיעה מעדכנים את הנתונים אחרי קריאה של מד המרחק. קריאות אלו נשמרות במערך מטיפוס מילה על מצביע המשתנה distances. כתבו תכנית מתאימה המחשבת את מספר הקילומטרים שהרכב נסע במשך החודש. דוגמה למערך המאחסן את פרטי הקריאות אשר בו האיבר הראשון מכיל את הקריאה של מד המרחק בתחילת החודש:

distances DW oA8gFh, oAFC6h, oFD8gh, 1oh, 12Fh, 345h

איבר אחרון במערך הוא 1-
שימו לב שמד המרחק מתאפס אם הקריאה עוברת את הערך FFFFh ואז הוא מתחיל את הקריאה מחדש מאפס.

תרגיל 12

כתבו תכנית שתגדיר מערך דו־ממדי מטיפוס בית בגודל 5x5 שיכיל מספרים מסומנים. התכנית תבצע את הפעולות הבאות:

- א. תחשב את מספר המספרים השליליים במערך ותשמור את התוצאה במשתנה countneg.
- ב. תבדוק כמה שורות במערך מכילות רק מספרים שליליים ותשמור את התוצאה במשתנה negrow.

תרגיל 13

במרוץ מרתון התחרו חמישה רצים בשלב הגמר. כל ספורטאי קיבל מספר מתחרה ואת התוצאה שלו בשניות. כתבו תכנית שבה יוגדרו חמש רשומות שיכילו את הפרטים הבאים:

- מספר רץ (בין 0 ל-10).
- תוצאות בשנייה (בין 10 ל-20).

התכנית תחשב, תימצא את הרץ המהיר ביותר ותשמור את החישוב במשתנה winner. הניחו כי תוצאות הריצה יהיו שונות זו מזו.

פרוצדורות ומקרו

חומר רקע

פעילות זו מתייחסת לפרק 7 בספר הלימוד



תרגיל 1

- נתון מערך בגודל N שכל איבר בו הוא מטיפוס בית.
 כתבו תכנית שתבדוק אם בין איברי המערך בשם `numbers` יופיע המספר 11.
 התכנית תשתמש בפרוצדורה שתקבל שני פרמטרים:
 א. הפרמטר ראשון יהיה פרמטר לפי כתובת שיצביע לתחילת המערך.
 ב. הפרמטר שני יהיה פרמטר לפי ערך שיציץ את מספר איברי המערך.

הפרוצדורה תבדוק ותאפס כל איבר במערך שערכו יהיה 11 (האיבר ישנה את תוכנו לאפס).

תרגיל 2

כתבו תכנית שתמנה את מספר הסיביות שערכן '1' במערך בשם `arr`. המערך `arr` יהיה מטיפוס מילה ובו יהיו N איברים. מספר הסיביות שערכן '1' יישמר במשתנה `numOfBits` מטיפוס מילה.

- התכנית תשתמש בפרוצדורה בשם `countDigit` שתקבל שני פרמטרים:
 א. פרמטר ראשון יהיה פרמטר לפי ערך והוא יהיה איבר במערך שיש לבדוק.
 ב. פרמטר שני יהיה פרמטר לפי כתובת המצביע למשתנה `numOfBits` שבו נצברים מספר הסיביות שערכן '1'.

תרגיל 3

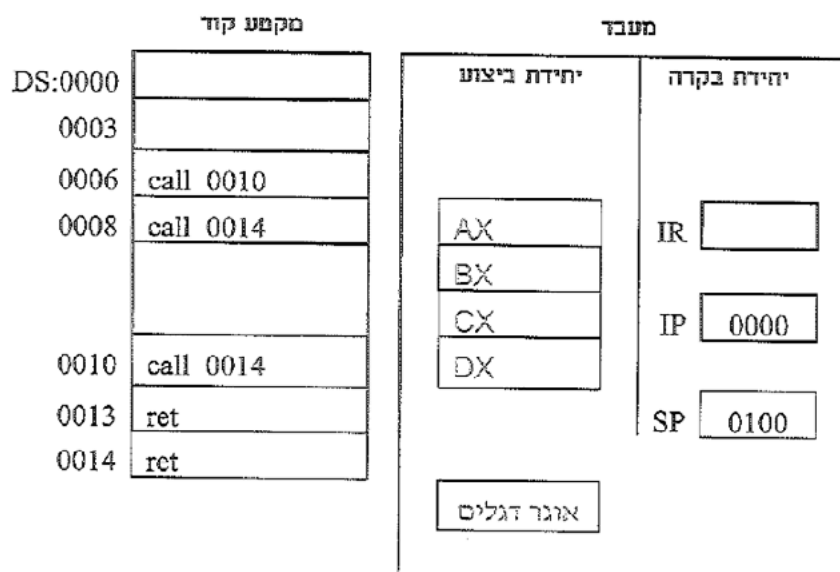
להלן תכנית המזמנת שתי פרוצדורות:

```
.MODEL SMALL
.STACK 100h
.DATA
.CODE
start:
    mov ax, @DATA
    mov ds, ax
    call A          ; 1
    call B          ; 2
    mov ax, 4C00h
    int 21h

    A PROC
    call B          ;3
    ret             ;4
A ENDP

B PROC
    Ret            ; 5
ENDP B
END start
```

השתמשו במפת הזיכרון להלן (המכילה קטע מהתכנית) כדי לתאר את מצב המחסנית לאחר כל אחת מההוראות הממוספרות ב:1 עד 5.



מקטע מחסנית	
SP→ SS:0100	
00FE	
00FC	
00FA	

תרגיל 4

בחברה שבה מכהנים 16 חברי הנהלה הוחלט להתקין מערכת הצבעה אלקטרונית. בכל הצבעה חבר הנהלה יוכל להצביע בעד או נגד. חבר הנהלה שיימנע מלהצביע ייחשב כמי שהצביע נגד ההצעה. הצעה תתקבל אם רוב מוחלט של חברי הנהלה יצביעו בעדה. מערכת ההצבעה האלקטרונית תקלוט את הקולות של חברי הנהלה ותשמור את המידע אם ההצעה שהועלתה התקבלה או לא.

כתבו תכנית בשפת אסמבלי שתבדוק האם ההצעה התקבלה. התכנית תגדיר משתנה votes מטיפוס מילה שבו כל סיבית תייצג הצבעה של חבר הנהלה אחד ומשתנה result מטיפוס בית שבו יישמרו הערכים 'ו' אם ההצבעה תתקבל ו-'ס' אם היא לא תתקבל.

א. הגדירו פונקציה שתקבל את ההצבעה להצעה כפרמטר לפי ערך. הפונקציה תחזיר באוגר AH :
 '1' - כשההצעה תתקבל
 '0' - כשההצעה לא תתקבל

ב. כתבו תכנית שתשתמש בפונקציה שהגדרתם בסעיף א'.

תרגיל 5

נתון מערך בשם arr. המערך arr הנו דו-ממדי $N \times N$ מטיפוס בית. כתבו תכנית שתבדוק אם סכום המספרים באלכסון הראשי (השמאלי) שווה לסכום המספרים באלכסון המשני (הימני). התכנית תשים במשתנה בשם diff את הערך '0' אם סכום שני האלכסונים שווה. אם סכומם אינו שווה, התכנית תשים במשתנה diff את ההפרש בין הסכום הגדול לסכום הקטן.

כדי לפתור את הבעיה כתבו שתי פרוצדורות:

- א. פרוצדורה ראשונה תחשב את הסכום של האלכסון הראשי.
- ב. פרוצדורה שנייה תחשב את הסכום של האלכסון המשני.

כל פרוצדורה תקבל כפרמטר לפי כתובת מצביע לאיבר הראשון של האלכסון המתאים וכפרמטר לפי ערך את N.

תרגיל 6

התכנית הבאה משתמשת בפרוצדורה בשם secret המקבלת שני פרמטרים:

- א. פרמטר ראשון לפי כתובת המצביע למחרוזת בשם number המכילה ספרות בין 0 ל-9. הניחו כי במחרוזת יש ספרה אחת לפחות שאינה 0.
- ב. פרמטר שני לפי כתובת המצביע למשתנה XX.

עקבו אחר ביצוע התכנית ורשמו מה יהיה ערכו של xx בסיום התכנית ומה תפקיד הפרוצדורה secret.

```

.MODEL SMALL
.STACK 100h
.DATA
    number DB "ooo456"
    xx DB ?
.CODE
start:
    mov ax, @DATA
    mov ds, ax
    lea bx, xx
    push bx
    lea bx, number
    push bx
    call secret
    mov ax, 4cooh
    int 21h
secret PROC
    mov bp sp
    mov bx, [bp+2]
    xor cx, cx
again:
    cmp byte ptr [bx], "o"
    jne finish
    inc cx
    inc bx
    jmp again
finish: mov bx, [bp +4]
    mov [bx], cx
    ret
secret ENDP
END start

```

תרגיל 7

זוג מספרים נקרא "מאורגן" אם זוג המספרים וסכומם הנם בעלי אותו סימן. כתבו תכנית שתבדוק את זוגות המספרים בגודל בית כל אחד. כל זוג מספרים יאוחסן במילה אשר בה מספר אחד יאוחסן בבית התחתון של המילה והמספר השני בבית העליון. זוגות המספרים יאוחסנו במערך בשם pairs. הזוג האחרון במערך הוא -1, -1.

- כתבו פונקציה שתקבל שני מספרים כשני פרמטרים ותחזיר '1' אם הם זוג תקין ו-'0' אם הם זוג לא תקין.
- כתבו תכנית שתשתמש בפונקציה שכתבתם בסעיף א' כדי לבדוק את מספר הזוגות התקינים במערך. התכנית תשמור במשתנה count את מספר הזוגות התקינים במערך.

תרגיל 8

כתבו תכנית שתמייין את איברי מערך בשם arr מטיפוס בית. אורך המערך יוגדר על ידי קבוע LEN.

תרגיל 9

נתונים שני מערכים firstAr ו secondAr. כל מערך מכיל מספרים בגודל מילה. שני המערכים מסתיימים ב-0. ידוע כי במערך firstAr כל האיברים שונים זה מזה ואילו במערך secondary ישנן חזרות של האיברים. כתבו תכנית שתבדוק ושתמנה לכל איבר במערך firstAr את מספר הפעמים שהוא מופיע במערך secondary. התכנית תאחסן את מספר ההופעות של כל מספר במערך שלישי בשם moneAr בהתאם למיקום המספר במערך firstAr.

לדוגמה:

שם מערך / מציין איבר						
0	1	2	3	4	5	6
3	1	7	0	2		
1	7	3	7	1	7	0
1	2	3	1	0		
					firstAr	
					secondAr	
					moneAr	

- א. הגדירו פונקציה שתקבל את הפרמטרים הבאים:
- פרמטר לפי כתובת ערך שהוא איבר במערך firstAr.
 - פרמטר לפי כתובת המצביע למערך secondAr.
- הפונקציה תחזיר את מספר ההופעות של האיבר במערך secondary.
- ב. השתמשו בפונקציה שהגדרתם בסעיף א' וכתבו תכנית מתאימה.

תרגיל 10

להלן תכנית המשתמשת במקרו:

- א. הסבירו מה מבצע המקרו ומה מבצעת התכנית עצמה.
- ב. מה תפקיד ההוראות push ax ו pop ax – הרשומות במקרו addMacro?

```
.MODEL SMALL
addMacro MACRO num1, num2, result
    push ax
    mov ax, num1
    add ax, num2
    mov result, ax
    pop ax
ENDM

.STACK 100h
.DATA
    a DW 29
    b DW 15
    c DW ?

.CODE
start:
```

אתחול מקטע הנתונים;

```
mov ax, @DATA  
mov ds, ax
```

קריאה למקרו;

```
addMacro a, b, c
```

יציאה מהתכנית;

```
mov ax, 4c00h  
int 21h  
END start
```

תרגיל 11

יש למצוא את המספר הגדול מבין שני מספרים ללא סימן מטיפוס בית.

- א. הגדירו מקרו שיגדיר שלושה פרמטרים: שני המספרים לבדיקה ואת המספר הגדול ביניהם.
- ב. כתבו תכנית שתשתמש במקרו שהגדרתם בסעיף א'.
- ג. שנו את המקרו ואת התכנית המשתמשת בו (שכתבתם בסעיפים א' ו-ב') כך, שיטפל במספרים עם סימן.

קלט פלט ופסיקות

חומר רקע

פעילות זו מתייחסת לפרק 8 בספר הלימוד



תרגיל 1

כתבו תכנית שתקלוט את שמו של המשתמש ותציג את השם עשר פעמים.

תרגיל 2

כתבו תכנית שתקלוט מהמשתמש שני מספרים מטיפוס מילה ותציג את סכומם ואת הפרשם בשתי שורות נפרדות. הקפידו להציג הודעות מתאימות שינחו את המשתמש בפעולות שעליו לבצע.

תרגיל 3

כתבו תכנית שתקלוט מהמשתמש מחרוזת של תווים. המחרוזת תכיל אותיות וספרות והתו האחרון שייקלט יהיה '\$'. התכנית תציג את מספר התווים במחרוזת ואת מספר האותיות בה.

תרגיל 4

הסבירו את המושג 'פסיקה' ופרטו אילו סוגים של פסיקות קיימות במעבד 8086.

מחרוזות

חומר רקע

פעילות זו מתייחסת לפרק 9 בספר הלימוד



תרגיל 1

מה עושה כל אחת מהשגרות הרשומות למטה?

א.

```
AProc  PROC
        Cld
        mov si, offset n1
        mov di , offset n2
        mov cx, 6
        rep movsw
        ret
AProc ENDP
```

ב.

```
CProc      PROC
           cld
           mov bx, 1
           mov ax, 0128h
           mov di, n1
           mov cx, 26
           repnz scasw
           jnz not_found
           mov bx, 1
not_found: ret
CProc      ENDP
EProc      PROC
           cld
           mov di, n3
           mov cx, 7
           mov ax, 3232h
           rep stows
           ret
EProc      ENDP
```

ג.

```
EProc      PROC
           cld
           mov di, n3
           mov cx, 7
           mov ax, 3232h
           rep stows
           ret
EProc      ENDP
```

תרגיל 2

ידוע כי המשתנה MyStr מכיל ספרות. מה מטרת התכנית הרשומה מטה ?

```
.MODEL      SMALL
.STACH      100h
.DATA
           mayStr db '1234.56'
```

```

        N db 7
        IsThere db 0

.CODE
start:   mov ax, @DATA
        mov ds, ax
        mov es, ax
        mov al, '.'
        mov di, offset MyStr
        xor cx, cx
        mov cl, n
        cld
        repnz scasb
        jne sof
        dec di
        mov bx, di
        mov di, offset N
        dec di
        std
        xor cx, cx
        mov cl, n
        repnz scasb
        inc di, bx
        jne sof
        inc IsThere
sof:     mov ah, 4ch
        int 21h
END      start

```

תרגיל 3

נתונה המחרוזת pattern המכילה ארבעה תווים. כתבו תכנית שתשרשר את המחרוזת חמש פעמים ותשים את התוצאה במשתנה newPattern.
לדוגמה: נתונה המחרוזת " \$\$@@ " . Pattern = " \$\$@@ " בסיום התכנית נקבל במשתנה newPattern את המחרוזת: " \$\$@@ \$\$@@ \$\$@@ \$\$@@ " .

השלימו את המקומות הריקים בקטע התכנית שלהלן:

.DATA

```
N EQU 4*5
Pattern DB '$$@@'
newPattern DB N dup (?)
```

.CODE

start:

```
mov ax, @DATA
mov ds, ax
move s, ax

-----
mov si, offset pattern
mov di, offset newPattern

-----
shr cx, 1

-----
mov ax, 4cooh
Int 21h
```

תרגיל 4

הגדירו משתנה בשם `str1` שיכיל מחרוזת באורך עשרה תווים. כתבו תכנית שתשים במשתנה `10ppStr` את המחרוזת שתורכב מתווי `str1` בסדר הפוך.

דוגמה: נתון `str1 = "ABCDE12345"`

התוצאה `10ppStr = "54321EDCBA"`

תרגיל 5

הגדירו משתנה בשם `text` שיכיל משפטים וכל משפט יכיל אותיות אנגליות קטנות ויסתיים בנקודה. כתבו תכנית שתשנה את האות הראשונה בכל משפט לאות אנגלית גדולה.

תרגיל 6

להלן תכנית שאמורה למחוק את כל תווי הרווח ממחרוזת `seq`. המשתנה `N` מכלל מספר המתאר את כמות התווים במחרוזת `seq`, התכנית אינה נכונה. תקנו אותה. שימו לבי המחרוזת `seq` יכולה להכיל רק את תו הרווח. במקרה זה צריך לעדכן את `N`.

```

.MODEL SMALL
.STACK 100h
.DATA
    N db 7
    Seq db "123 84"
.CODE
start:
    mov ax, @data
    mov ds, ax
    xor cx, cx
    mov cl, n
    mov al, ' '
    cld
Again    mov di, offset seq
        repnz scasb
        call moveBack
        jz again
        mov ah, 4ch
        int 21h
moveBack PROC
again1:  mov ah, [di + 1]
        mov [di], ah
        loop again1
        dec N
moveBack ENDP
END Start

```

תרגיל 7

הגדירו משתנה בשם `name` שבו תאוחסן מחרוזת שתכיל אותיות אנגליות גדולות. כתבו תכנית שתבדוק אם הצירוף "AB" נמצא במחרוזת. אם כן, שימו במשתנה בשם `startName` את התווים הנמצאים במחרוזת הנתונה לפני הצירוף "AB". אחרת שימו במשתנה `startName` את התו רווח.

תרגיל 8

הגדירו משתנה בשם `str1` שיכיל מחרוזת באורך `N`. המחרוזת תוכל להכיל אותיות אנגליות וספרות. כתבו תכנית שתבדוק אם המחרוזת מכילה רק אותיות אנגליות. אם המחרוזת אינה מכילה ספרות יש לשים במשתנה `check` את התו "!", אחרת, יש לשים במשתנה את מיקום הספרה הראשונה שנמצאה במחרוזת.

תרגיל 9

הגדירו משתנה בשם `str` שיכיל מחרוזת (מטיפוס בית) באורך `N`. כתבו תכנית שתבדוק אם המחרוזת מכילה רק ספרות בין 0 ל-9. אם כן, התכנית תחבר את הספרות ותשים את התוצאה במשתנה בשם `result`. אחרת, התכנית תשים במשתנה `result` את הערך '1-'.
לדוגמה:

נתונה המחרוזת "132054898" - `str`
בסיום התכנית ערכו של `result` הוא '40'.

מעבדים מתקדמים

חומר רקע

פעילות זו מתייחסת לפרק 10 בספר הלימוד



תרגיל 1

תארו כיצד ההתפתחות הטכנולוגית השפיעה על ביצועי המעבד.

תרגיל 2

ציינו את ההשפעות של ההתפתחות הטכנולוגית על מעבדים במשפחת אינטל. התייחסו להשפעה של מהירות השעון, לרוחב פס הנתונים ולרוחב פס הכתובות.

תרגיל 3

מה גודל הזיכרון של מעבד שבו פס הכתובות הוא בן 64 סיביות? ציינו גודל זה Gbyte.

תרגיל 4

כתבו קטע של תכנית שיסכם תוכן שלושה משתנים מטיפוס מילה כפולה ויאחסן את התוצאה במשתנה רביעי מטיפוס מילה כפולה. השתמשו באוגרים של מעבד פנטיום ובהוראות ADD, MOV, ADC.

תרגיל 5

הגדירו את המושג "ארכיטקטורת צינור ההוראות".

תרגיל 6

חשבו זמן לטיפול בעשרה לקוחות אם:

א. בכל לקוח מטפלים ארבעה עובדים הפועלים בשיטת "צינור הוראות". זמן הפעולה של כל עובד הוא שתי יחידות זמן.

ב. בכל לקוח מטפלים ארבעה עובדים הפועלים בשיטת "צינור הוראות". זמן הפעולה של כל עובד הוא יחידת זמן אחת.

תרגיל 7

הסבירו את התפקיד של תור ההוראות. מנו שלושה מקרים שבהם יחידת הביצוע צריכה להמתין להבאת הוראה.

תרגיל 8

- א. הסבירו מדוע השימוש בחוצצים עשוי להאריך את זמן הביצוע של הוראה.
ב. הסבירו אם הטענה הבאה נכונה: מאחר שהשימוש בחוצצים מאריך את זמן הביצוע של הוראה, לא כדאי להשתמש בחוצצים בארכיטקטורה של צינור הוראות, וכל יחידה תמסור מידע ישירות ליחידה העוקבת לה.

תרגיל 9

הסבירו את המושג סיכון (hazard) ומנו שלושה מצבי הסיכון.

תרגיל 10

התבוננו בקטע הקוד הבא ורשמו האם יש מצב סיכון הנגרם ממבנה (hazard structure):

א. במחזור מכונה מספר שתיים.

ב. במחזור מכונה מספר ארבע.

נמקו.

	1	2	3	4	5	6	מספר מחזור מכונה
mov ax ,bx	F	D	E	w			
mov var12 ,h		F	D	E	w		
add bx ,var			F	D	E	w	

תרגיל 11

התבוננו בקטע הקוד שלהלן ורשמו אם יש בו מצב של סיכון נתונים (data hazard) אם כן, באילו מחזורי מכונה? נמקו.

	1	2	3	4	5	6	מספר מחזור מכונה
mov ax ,bx	F	D	E	w			
mov var12 ,h		F	D	E	w		
add bx ,var			F	D	E	w	

תרגיל 12

השתמשו בהשהיה כדי לפתור בעיות אלו. אם כל מחזור מכונה הוא יחידת זמן אחת כמה זמן יארך ביצוע הוראות אלה?

תרגיל 13

כיצד שימוש בשיטת "גניבת ערכים" (forwarding data) פותר את בעיית סיכון נתונים (hazard data)?

תרגיל 14

תארו שתי שיטות שבהן אפשר לפתור את בעיית סיכון נתונים (hazard data) בתוכנה.

הקטע הראשון

DS:0000

0003

0005

0008

000B

000C

000D

0010

0011

0013

001C

001C

CMP BX, 0

JG 000B

MOV AX, 1

JMP 0013

PUSH BX

DEC BX

IR MOV AX, 4c00h

IP POP BX

SP EMUL BX

RET

הקטע השני



R

הקטע השלישי

הקטע הרביעי

PUSH 3

factorial(3) - מחשב את הערך של פונקציית פקטוריאל

PUSH 2

factorial(2) - מחשב את הערך של פונקציית פקטוריאל

PUSH 1

factorial(1) - מחשב את הערך של פונקציית פקטוריאל

00FA

00F8

00F6

00F4

00F2

00F0

0010B

0002

0010B

0001

0010

AX = 6

BX = 3

CX

DX

002:7

AX = 6

BX = 3

CX = 001F

DX = 0100

002:7



78-1043312 דאנאקוד