

```
mov [1], ax
```

שקולה לפקודות הבאות:

```
mov bx, 1
```

```
mov [bx], ax
```

בדוגמה הרביעית, ערכו של ax יועתק לתוך הזיכרון לכתובת השמורה ב-bx ועוד 1, כלומר בית אחד אחרי הכתובת השמורה ב-bx.

תרגיל 6.6



א. הגדירו בתוך DATASEG את המשתנה var בגודל בית, קבעו את ערכו ההתחלתי בתור 0. העתיקו לתוך al את הערך 100 (דצימלי) ולתוך bx את הערך 2. הוסיפו את שורות הקוד הבאות לתוכנית:

```
mov [Var], al
```

```
mov [1], al
```

```
mov [bx], al
```

```
mov [bx+1], al
```

ב. הריצו את התוכנית ב-TD ועיקבו אחרי השינויים ב-DATASEG בזמן ריצת התוכנית. וודאו שכל ארבעת הבתים הראשונים ב-DATASEG מקבלים את הערך 100 (כייצוג ההקסדצימלי שלו, כמובן).

העתקה של תא בזיכרון אל רגיסטר

השיטות להעתיק תא בזיכרון לתוך רגיסטר הן בדיוק כמו השיטות להעתיק רגיסטר לתוך תא בזיכרון, בהבדל אחד – האופרנדים הפוכים:

```
mov register, memory
```

לכן, כל הדוגמאות שנתנו בסעיף הקודם תקפות, רק בהיפוך אופרנדים:

```
mov ax, [1]
```

```
mov ax, [Var]
```

```
mov ax, [bx]
```

```
mov ax, [bx+2]
```

תרגיל 6.7



בהמשך לתרגיל הקודם, הוסיפו לתוכנית את שורות הקוד הבאות (אחרי שורות הקוד של התרגיל הקודם):

```

mov  [Var], al
mov  [1], al
mov  [bx], al
mov  [bx+1], al
;—1—
mov  al, 0
mov  al, [var]
;—2—
mov  al, 0
mov  al, [1]
;—3—
mov  al, 0
mov  al, [bx]
;—4—
mov  al, 0
mov  al, [bx+1]

```

הריצו את התוכנית ב־TD ועיקבו אחרי השינויים ב־al בזמן ריצת התוכנית. ודאו שבכל אחד מארבעת הקטעים al מקבל את הערך 100 (בייצוג ההקסדצימלי שלו, כמובן).

- ▶ צרו תוכנית, שמגדירה בתוך DATASEG מערך של בתים, שגודלו מספיק על מנת לשמור את השם שלכם. הגדרת המערך לא תכלול ערכים.
- ▶ מיד לאחר הגדרת המערך, הגדירו מחרוזת של בתים. הגדרת המחרוזת תכלול את מספר הטלפון שלכם.
- ▶ בתוך CODESEG כיתבו קוד שמעתיק לתוך המערך את השם שלכם, אות אחרי אות