# פקודות בקרה - תנאים לוגיים

## cmp

**צורת כתיבה חוקית של פקודת cmp**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| תוצאה |  | דוגמה |  | הפקודה |
| שינוי מצב הדגלים בהתאם  ליחס בין האופרנדים | cmp | al, bl | cmp | register, register |
| cmp | al, [byteVar] | cmp | register, [memory] |
| cmp | [WordVar], cx | cmp | [memory], register |
| cmp | ax, 5 | cmp | register, constant |
| cmp | [WordVar], 5 | cmp | [memory], constant |

**ריכוז פקודות jmp**

|  |  |  |
| --- | --- | --- |
| כל פקודת jmp תחיל ב – J  N - not / E - equal  משמעות הפקודה | כל פקודת jmp תחיל ב – J  G - greater / L - less  מספרים Signed | כל פקודת jmp תחיל ב – J  A – above / B - below  מספרים Unsigmed |
| קפוץ אם אופרנד יעד גדול מהמקור | JG – Jump if Greater | JA – Jump if above |
| קפוץ אם אופרנד יעד קטן מהמקור | JL – Jump if Less | JB – Jump if Below |
| קפוץ אם אופרנד יעד ומקור שווים | JE – Jump Equal | |
| קפוץ אם אופרנד יעד ומקור שונים | JNE – Jump Not Equal | |
| קפוץ אם אופרנד יעד גדול או שווה לאופרנד המקור | JGE – Jump if Greater or Equal | JAE – Jump if Above or Equal |
| קפוץ אם אופרנד יעד קטן או שווה לאופרנד המקור | JLE – Jump if Less or Equal | JBE – Jump if Below or Equal |

**בדיקה האם הקשנו על מקש q:**

|  |  |
| --- | --- |
| הפסיקה של המתנה למקש שומרת את ערך המקש לריגסטר al.  ההוראה cmp משווה האם הערך השמור ברגיסטר al = לאות q  je אם הערך שווה (=)  קופצים לתווית end\_game:  ממשיכים בפקודות על פי הסדר הרגיל. | ; waits for character   * mov ah, 0h * int 16h   ; check if user asks to quit   * cmp al, 'q' * je end\_game   . . . . . . .  end\_game:  . . . . . . . |

שמרו את התכנית שיצרתם בשיעור הקודם בשם חדש והמשיכו לעבוד בקובץ החדש.

## האם לצאת מהתכנית Quit?

בתכנית שכתבנו בשיעור קודם הזזנו את השחקן על המסך, בעזרת פקודות חיבור וחיסר.

לפני הפסיקה המחזירה את המסך לתצוגת טקסט ויציאה מהתכנית, הוסיפו תווית

end\_game:

end\_game:

; text mode

mov ax, 2h

int 10h

חזרו מסוף הקוד אחורה ל “wait for character” שלפני התזוזה האחרונה של השחקן – לפני התנועה באלכסון

נוסיף לפני מקבץ פקודת אלו את התנאי שיבדוק מהו ערך המקש שהקשנו לאחר הפסיקה של המתנה למקש:

; waits for character

mov ah, 0h

int 16h

; check if user asks to quit

cmp al, 'q'

je end\_game

שנו את סט הפקודות של תזוזת השחקן **אחת לפני הסוף**  לבדיקה אם הקשנו על מקש 'q'

שנו את צבע השחקן בסט פקודות אלו את צבע השחקן ללבן.

mov [color], 15

שמרו והריצו.

בדקו את ההרצה פעם אחת, כאשר אתם רואים את השחקן הלבן, הקישו על מקש q ⇦ יציאה מהתכנית.

בדקו שוב את ההרצה, כאשר אתם רואים את השחקן הלבן, הקישו על כל מקש אחר ⇦ התכנית מציגה עוד תזוזה של השחקן, ממתינים למקש ויוצאים מהתכנית.

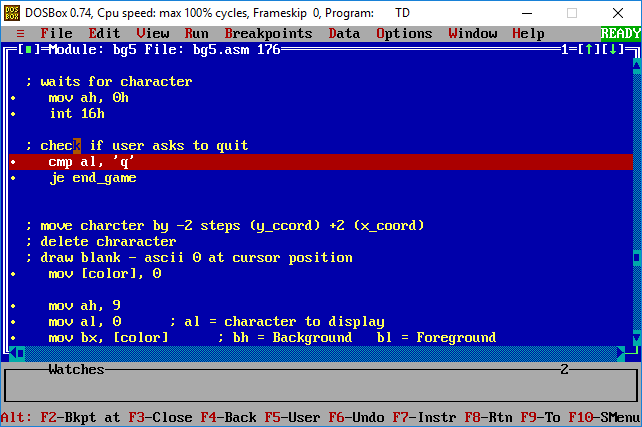
הריצו שוב את התכנית עם ה – TD,

רדו עם הסמן לשורה שבה רשומה פקודת ההשוואה.

cmp al, 'q'

בחרו עם העכבר בתפריט העליון במילה Breakpoints

השורה תהפוך לשורה אדומה.



זהו סימן לתכנית לעצור בהרצה בפקודה זו.

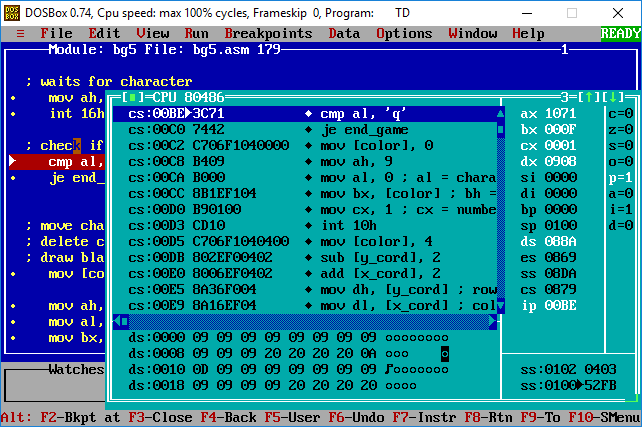
נריץ את התכנית עם F9.

כאשר נגיע לשחקן הלבן נקיש על q במקלדת

נחזור למסך ה – TD

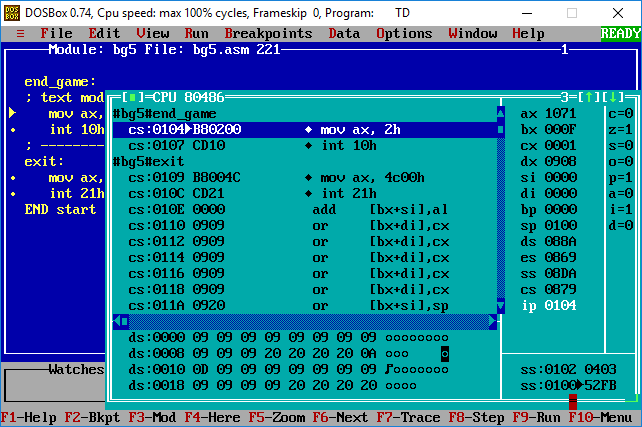
נצפנה בתצוגת המעבד, Alt + v ⇨ C

שימו לב לרגיסטר ה – IP ⇦ הוא קיבל את הערך של הפקודה לביצוע (פקודת ההשוואה)



נריץ עת הפקודה ע"י F8 ⇦ נעבור לפקודת הקפיצה ורגיסטר IP קפץ בשני ערכים

נריץ את פקודת הקפיצה ⇦ שימו לב לרגיסטר IP הוא קיבל את הערך של הפקודה העוקבת למיקום התווית end\_game



## קפיצה רחוקה

עכשיו ננסה להוסיף את הפקודה של אפשרות יציאה מהתכנית בתחילת המשחק.

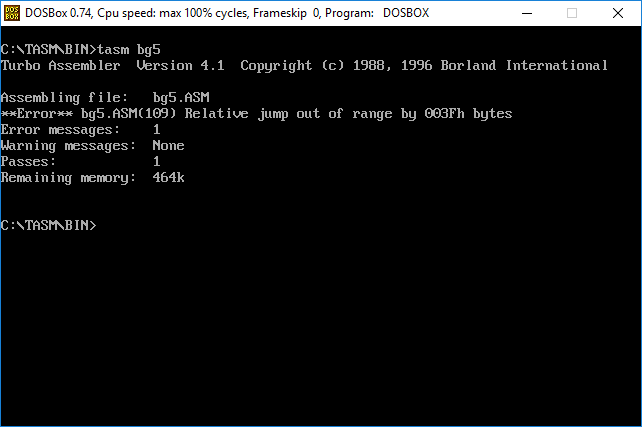
**אחרי ציור לוח המשחק וההופעה הראשונה של השחקן.**

הוסיפו לאחר הפסיקה של “wait for character” את הבדיקה האם הקשנו על מקש q?

שמרו והריצו.

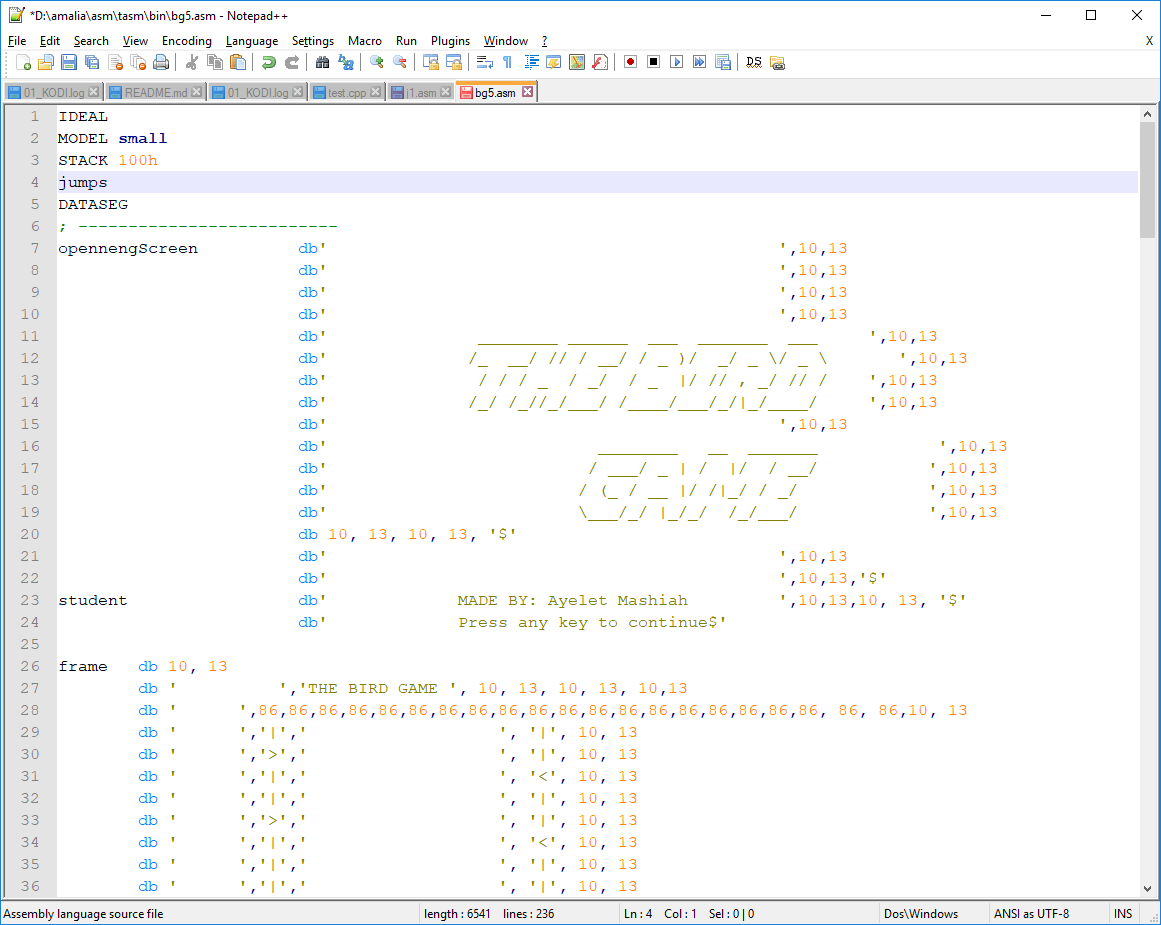
בהרצה של tasm כאשר הופכים את הקובץ לשפת מכונה, קיבלו הודעת שגיאה.

Relative jump out of range



הודעה זו אומרת לנו שהקפיצה שהאסמבלר צריך לעשות היא של שורות רבות מידי.

כדי למנוע בעיה זו נוסיף בתחילת הקובץ את הפקודה jmps המאפשרת לאסמבלי ליצור קפיצות גדולות.



הוסיפו את הפקודה

הריצו את התכנית.

