# פעולות אריתמטיות לוגיות

## חיבור וחיסור

הזזת השחקן על פני לוח המשחק.

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

בשיעור זה נתרגל את התזוזה של השחקן שלנו.

לצורך תזוזת הדמות יצרנו משתנים אשר ישמרו את הערכים של מיקום הסמן על ציר x, מיקום הסמן על ציר y וצבע הדמות.

במקטע הנתונים – DATASEG

; character coordinates

x\_cord db 7 ; column

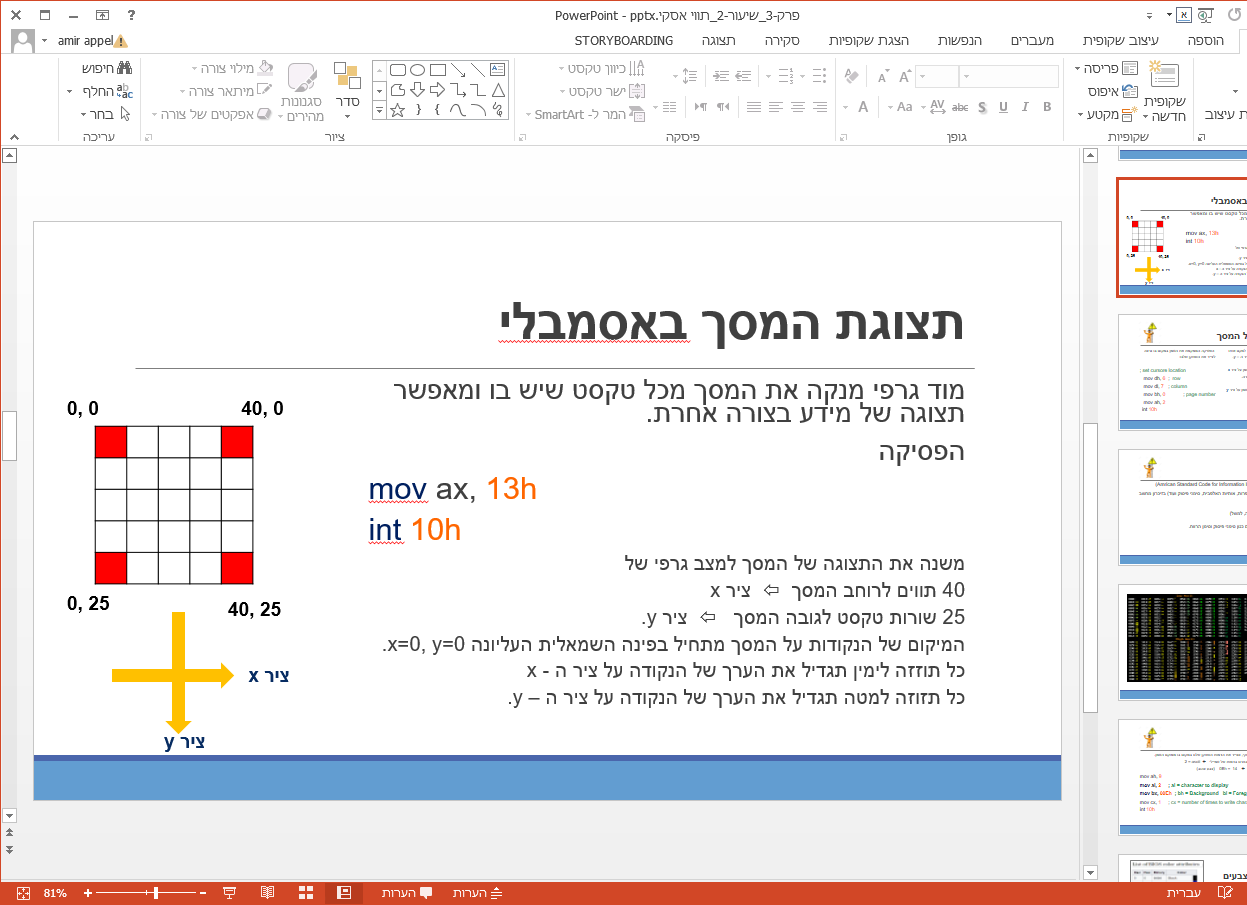
y\_cord db 6 ; row

color dw 0Eh ; color

למדנו שכדי שנזיז את הדמות (ליצור אנימציה) "מוחקים" את הדמות ע"י ציור הדמות בצבע שחור במיקום בו היא נמצאת.

שינוי מיקום הסמן על ציר ה – x ועל צירה ה – y

ציור השחקן בצבע במיקום החדש.

כדי להזיז את הדמות ימינה יש להגדיל את המיקום על ציר ה – x ⇦

כדי להזיז את הדמות שמאלה יש להקטין את המיקום על ציר ה – x ⇨

כדי להזיז את הדמות למטה יש להגדיל את המיקום על ציר ה – y ⇩

כדי להזיז את הדמות למעלה יש להקטין את המיקום על ציר ה – y ⇧

הפסיקות הממקמות את הסמן ומציירות את השחקן.

;============ On key press, move character ===========

mov [color], 0 ; black color

"מחיקת הדמות" ציור הדמות בשחור במיקום הסמן .

; draw smiley - ascii 2 at cursor position

mov ah, 9

mov al, 2 ; aL = character to display

mov bx, [color] ; bh = Background bl = Foreground

mov cx, 1 ; cx = number of times to write character

int 10h

mov [y\_cord], 19

השמת ערכים חדשים לציור הדמות החדשה במיקום ובצבע חדש.

mov [x\_cord] , 5

mov [color], 4 ; red color

; print chararcter

; set cursore location

mov dh, [y\_cord] ; row

הזזת הסמן למיקום החדש וציור הדמות במיקום החדש

mov dl, [x\_cord] ; column

mov bh, 0 ; page number

mov ah, 2

int 10h

; draw smiley - ascii 2 at cursor position

mov ah, 9

mov al, 2 ; aL = character to display

mov bx, [color] ; bh = Background bl = Foreground

mov cx, 1 ; cx = number of times to write character

int 10h

; waits for character

המתנה למקש

mov ah, 0h

int 16h

;===============================================

## עבודה עם ה –Turbo Debugger

הריצו את התכנית ב TD

לפני תחילת ההרצה נבדוק את ערכי המשתנים:

Alt + v

נבחר ב – V

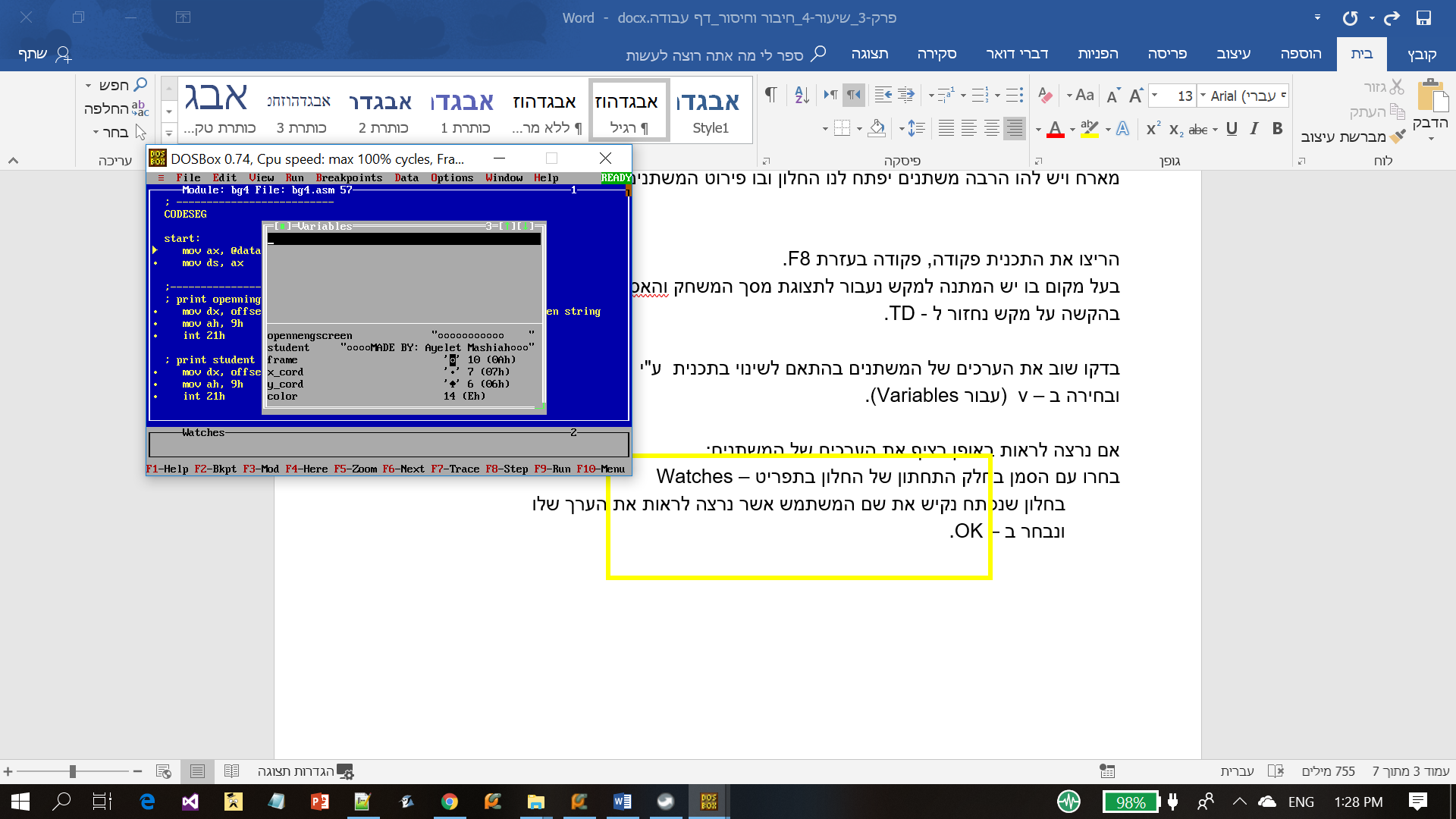
יפתח לנו החלון ובו פירוט המשתנים שלנו.

הריצו את התכנית פקודה, פקודה בעזרת F8.

בעל מקום בו יש המתנה למקש נעבור לתצוגת מסך המשחק והאסמבלי ימתין עד שנקיש על מקש ⇦ בהקשה על מקש נחזור ל - TD.

בדקו שוב את הערכים של המשתנים בהתאם לשינוי בתכנית ע"י Alt + v

ובחירה ב – v (עבור Variables).

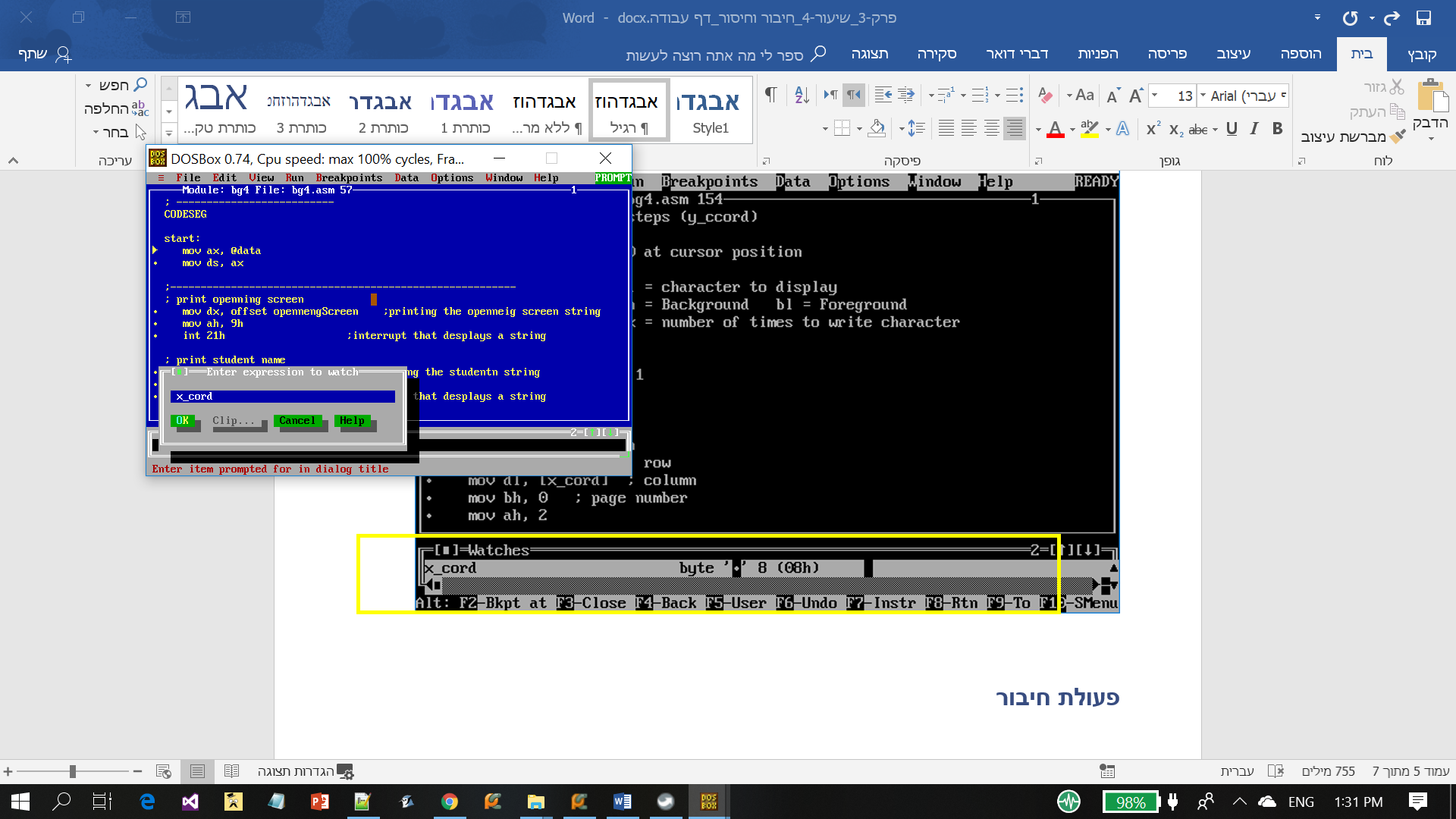


אם נרצה לראות באופן רציף את הערכים של המשתנים:

בחרו עם הסמן בחלק התחתון של החלון בתפריט – Watches

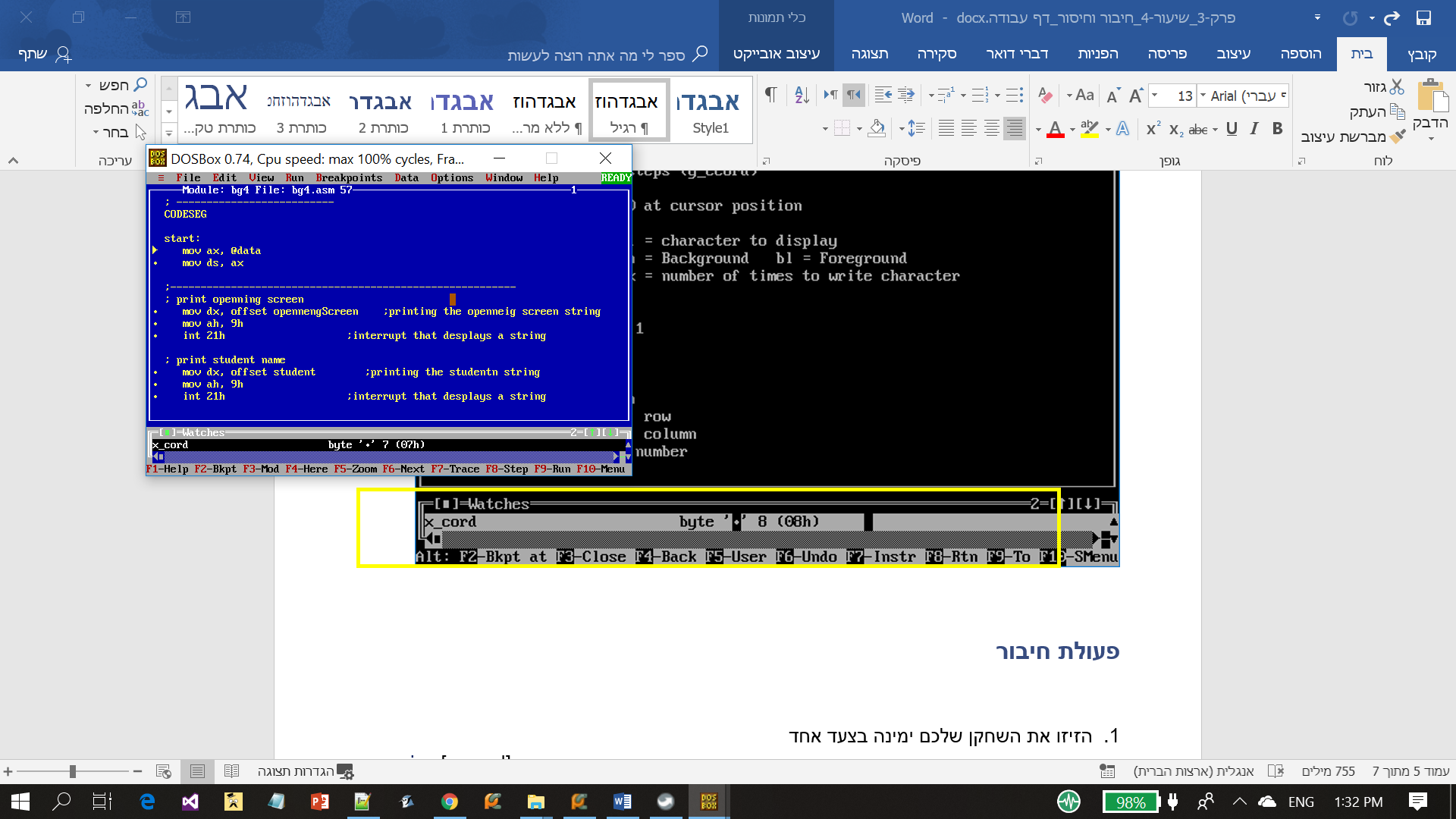
בחלון שנפתח נקיש את שם המשתמש אשר נרצה לראות את הערך שלו

ונבחר ב – OK.



נוכל לראות את הערך של המשתנה בחלון של ה – TD.

שם המשתנה גודלו ערכו בבסיס 10 (וערכו בבסיס הקסדצימאלי)



|  |
| --- |
| בשיעור הקודם הזזנו את השחקן מספר פעמים כל ידי פעולות השמה.  הפעם נזיז את הדמות שלנו על ידי **חישוב** הקואורדינטות של המיקום הבא ולא על ידי קביעתן.  נשנה את פקודות ההשמה לפקודות אריתמטיות לוגיות.  מחקו את פעולות ההשמה למשתנים [x\_cord] ן – [y\_cord] ובמקומם נרשום פקודות חיבור וחיסור.  אם נרצה לזוז ימינה נגדיל את x\_cord על ידי פעולת חיבור ואם נרצה לזוז שמאלה נקטין אות על ידי פעולת חיסור.  אם נרצה לזוז למטה נגדיל את y\_cord על ידי פעולת חיבור  ואם נרצה לזוז למעלה נקטין אות על ידי פעולת חיסור. |

## פעולת חיבור

1. הזיזו את השחקן שלכם ימינה בצעד אחד

inc [x\_cord]

1. הזיזו את השחקן שלכם ימינה ב - 3 צעדים

add [x\_cord], 3

1. הזיזו את השחקן שלכם למטה בצעד אחד

inc [y\_cord]

1. הזיזו את השחקן שלכם למטה ב - 5 צעדים

add [y\_cord], 5

## פעולת חיסור

1. הזיזו את השחקן שלכם שמאלה בצעד אחד

dec [x\_cord]

1. הזיזו את השחקן שלכם שמאלה ב - 3 צעדים

sub [x\_cord], 3

1. הזיזו את השחקן שלכם למעלה בצעד אחד

dec [y\_cord]

1. הזיזו את השחקן שלכם למעלה ב - 5 צעדים

sub [y\_cord], 5

1. האם תוכלו להזיז את השחקן שלכם באלכסון?

; move charcter by +2 steps (y\_ccord) +1 (x\_coord)

; delete chraracter

; draw blank - ascii 0 at cursor position

mov [color], 0

mov ah, 9

mov al, 0 ; al = character to display

mov bx, [color] ; bh = Background bl = Foreground

mov cx, 1 ; cx = number of times to write character

int 10h

; increase x\_coord by 1

; decrease y\_coord by 2

mov [color], 2

add [y\_cord], 2

inc [x\_cord]

; print chararcter

; set cursore location

mov dh, [y\_cord] ; row

mov dl, [x\_cord] ; column

mov bh, 0 ; page number

mov ah, 2

int 10h

; draw smiley - ascii 2 at cursor position

mov ah, 9

mov al, 2 ; aL = character to display

mov bx, [color] ; bh = Background bl = Foreground

mov cx, 1 ; cx = number of times to write character

int 10h

; waits for character

mov ah, 0h

int 16h