

* שם ביה"ס : תיכון הדרים, הוד השרון .

* נושא העבודה : ניחוש מספרים אקראיים ע"פ הסדר הנכון.
* שם מגיש העבודה : אופיר בן שוהם.
* ת.ז – 208642496 .

**פרק ראשון**

**שם הפרויקט : הגרלת מספרים אקראיים וניחושם**

**ע"פ הסדר הנכון.**



**פרק שני – נושא הפרויקט**

פרויקט זה, עוסק בהגרלת מספרים אקראיים וניחושם,תוך כדי מתן חשיבות לסדר שהמספרים מוגרלים.

בתחילת התוכנית ישנה הודעת פלט למשתמש: " הכנס 5 מספרים בין 1-9,כולל ". לאחר שהוא מכניס אותם, הם מסודרים בטבלה, שאותה ניתן לראות בפרק מספר 6: "אלגוריתם הפרויקט" .

מיד לאחר מכן מתבצעת הגרלת 5 מספרים אקראיים בטווח 1-9. אולם, בAssembly,בניגוד לשפות תכנות אחרות – אין פעולת random מובנת. לכן, בניתי אותה בעצמי. האלגוריתם של הפעולה שמגרילה מספרים אקראיים בפרויקט שלי עובדת כך: ההגרלה נעשית באמצעות מאיות השנייה של השעון במחשב. ההגרלה מתבצעת פעם ב55 MS. ההגרלה היא בעצם בין 0-100- כאמור, מאיות השנייה. אבל, אני צמצמתי את טווח ההגרלה ל0-9, ועשיתי זאת באמצעות אלגוריתם שמוסבר בתיעוד, בפרויקט עצמו.

לאחר הגרלת המספרים האקראיים, גם הם מסודרים בטבלה.

לאחר מכן, מופיעה טבלה נוספת שבודקתהאם המספר הראשון שהמשתמש הכניס כקלט זהה למספר הראשון שהוגרל, וכך הלאה גם לשאר המספרים.

\* לכאורה,נראה שבטווח מספרים קטן בין 1-9 ההסתברות להצלחת 5 הניחושים היא גבוהה. אולם, ההסתברות שהמשתמש ינחש נכונה את חמשת המספרים במיקום הנכון, היא 1 חלקי תשע בחזקת חמש = 1/59,049 .

בסיום התוכנית, ישנה טבלה המסכמת את ניחושי המספרים נכונה בהתאם למיקום שהוגרלו. ישנן שתי אפשרויות להצגה הערך בטבלה:

T (TRUE) OR F (FALSE) .

T – מוצג כאשר הקלט שהמשתמש הכניס זהה למספר שהוגרל אקראית.

F – מוצג כאשר הקלט שהמשתמש הכניס שונה מהמספר שהוגרל אקראית.

**פרק שלישי - סביבת עבודה**

סביבת העבודה שבה בחרתי לעשות את הפרוייקט באסמבלר הינה

EMU8086.

**פרק רביעי – סיבת בחירתי בנושא**

בחרתי את הנושא "ניחוש מספרים אקראיים" מכמה סיבות.

ראשית, רציתי ללמוד באופן עצמאי על הדרך להגריל מספרים אקראיים באסמבלר. גיליתי שהפעולה אינה פשוטה כמו בשפות עיליות, דבר שאתגר אותי ללמוד עוד בנושא.

שנית, השימוש במערכים באסמבלר עניין אותי. שמרתי במערכים את הקלט והפלט של התוכנית.

כמו כן, רציתי לשלב את המתמטיקה במדעי המחשב. ולכן, בדקתי מבחינה הסתברותית מה הסיכוי שהמשמש ינחש נכונה את כל חמשת המספרים.

**פרק חמישי- הוראות הפעלה**

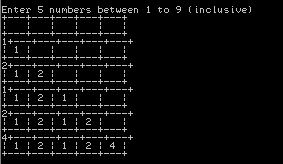
לאחר הרצת התוכנית, באמצעות כפתור הrun מוצגות במהלך התוכנית הודעות פלט למשתמש, שמציגות את המבוקש ממנו כדי שהתוכנית תמשיך. למשל, בתחילת התוכנית על המשתמש להכניס 5 מספרים. לאחר מכן התוכנית תמשיך לרוץ לבדה, ותציג במהלכה הודעות פלט שמסבירות למשתמש על המתרחש בה.

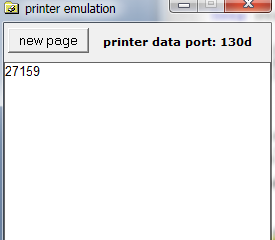
**פרק שישי : אלגוריתם הפרויקט**

בתחילת התוכנית, מוצגת למשתמש הודעה "enter 5 numbers",ומצוירת טבלה. כפי שניתן לראותה בצילום מסך הבא:

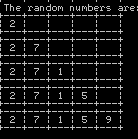


לאחר מכן, כאשר המשתמש מכניס 5 מספרים, התוכנית שומרת אותם במערך ששמו arr . בעת הכנסת הערכים התוכנית נראית כך:



אחרי הכנסת הקלט, התוכנית מגרילה 5 מספרים אקראיים.כל פעם שמוגרל ערך בטווח 0-10 (לא כולל), הוא מופיע בחלון כזה: 

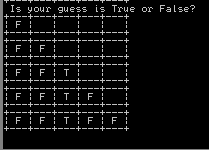
לאחר שמוגרלים 5 מספרים אקראיים, הם הנשמרים במערך ששמו arr2, הערכים האקראיים שהוגרלו מוצגים גם כן בטבלה,יחד עם הודעת פלט:



כעת יש לנו מערך arr (ובו הניחושים של המשתמש), ומערך arr2

( ובו ההגרלות האקראיות של המחשב). כעת באמצעות קטע הקוד שיצרתי אני משווה בין שני המערכים הנתונים. במידה ו arr [si] שווה לarr [si] , נשמר במערך arrcheck T . אחרת, הם לא שווים ונשמר F. חשוב לציין שsi מתקדם עד סופו של גודל המערך, כל פעם במקום אחד כדי לעבור על כל האיברים במערכים .

המערך arrcheck מציין האם האיבר במקום מסוים שהוכנס כקלט שווה לאיבר שהוגרל אקראית באותו מקום. לאחר ההשוואה, המערך הנתון מוצג בטבלה מסכמת:



כדי להוכיח את אמיתות הטבלה, אציג כעת את המערכים במלואם אחד מתחת לשני, ומתחתם טבלת ההשוואה ביניהם:

1. מערך arr - המספרים שהוכנסו כקלט:



2. מערך arr2 - המספרים שהוגרלו אקראית:



3. מערך arrcheck- בודק שיוון האיברים במערכים בסעיפים 1,2:



כדי לפשט את התוכנית, אתן דוגמה מוחשית:

האיבר במקום arr[0] הוא 1 , האיבר במקום arr2[0]הוא 2. 1 שונה מ2, ולכן במערך arrcheck[0] מוצג F .

אתאר את אלגוריתם המערכת בתרשים זרימה:

הכנסת 5 מספרים בין 1-9

הכנס לarrcheck[si] את התו F

הכנס לarrcheck[si] את התו T

'T'

בדיקה האם arr[si] שווה לarr2[si]

קידום si באחד .

שמירתם במערך arr2

הגרלת 5 מספרים אקראיים בין

1-9

שמירתם במערך arr

הצגת מערך arrcheck[si] בטבלה, תוך כדי קידום si באחד.

תיאור המשתנים בזיכרון

הודעות הפלט למשתמש, שמורות בזיכרון כך:

pkey db "press any key…"$

asktoinputdata db "Enter 5 numbers between 1 to 9 (inclusive)",0ah,0dh,"$"

db mone 5

showrandomnumbersa db " The random numbers are: " ,0ah,0dh,"$"

checkyourguess db " Is your guess is True or False? " ,0ah,0dh,"$"

מערכים מוגדרים, שמורים בזיכרון כך:

\* כל מערך מוגדר בגודל 5 .

arr db 5 dup (?) ; set arr for 5 random numbers

arr2 db 5 dup (?) ; set another arr for 5 input numbers

arrcheck db 5 dup (?) ; check if they equel or not

הגדרת הודעת הstart שמוצגת למשתמש .

הגדרת הטבלאות בזיכרון , כל טבלה מיועדת למערך אחר.

;

myBoard db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

myBoard2 db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

myBoard3 db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

הפרוצדורות:

push bp

mov bp,sp

push cx ; save before the change

lea dx, myBoard

mov ah, 09h

int 21h

pop cx ; Returns the value that was set before the change

pop bp ; returns bp before the changes

ret ; add ret beacuse i dont want again random numbers.

endp drawBoard ; end p. draw Board.

proc drawBoard2 ; draw Board .

push bp

mov bp,sp

push cx

lea dx, myBoard2

mov ah, 09h

int 21h

pop cx

pop bp

ret ; add ret beacuse i dont want again random numbers.

endp drawBoard2 ; end p. draw Board.

proc drawBoard3 ; draw Board .

push bp

mov bp,sp

push cx

lea dx, myBoard3

mov ah, 09h

int 21h

pop cx

pop bp

ret ; Returns to the place at the beginning of the procedure

endp drawBoard3 ; end p. draw Board.

קטע הקוד בשלמותו :

; multi-segment executable file template.

data segment

; add your data here!

pkey db "press any key...$"

asktoinputdata db "Enter 5 numbers between 1 to 9 (inclusive)",0ah,0dh,"$"

showrandomnumbersa db " The random numbers are: " ,0ah,0dh,"$"

checkyourguess db " Is your guess is True or False? " ,0ah,0dh,"$"

; i need to save the registers

; i need to save the input in other registers and compare between them

arr db 5 dup (?) ; set arr for 5 random numbers

arr2 db 5 dup (?) ; set another arr for 5 input numbers

arrcheck db 5 dup (?) ; check if they equel or not.

mygrapics db ' ' ,0ah,0dh

db ' ' ,0ah,0dh

db ' ' ,0ah,0dh

db ' | | | | ' ,0ah,0dh

db ' \_\_\_| |\_ \_\_ \_ \_ |\_ ' ,0ah,0dh

db ' / \_\_| \_\_/ | \_\_|| ',0ah,0dh

db ' \\_\_ \ || (\_| | | |\_ ' ,0ah,0dh

db ' |\_\_\_/\\_\_\\_\_,\_| \\_\_| ' ,0ah,0dh

db ' ' ,0ah,0dh

db ' ' ,0ah,0dh

db ' ' ,0ah,0dh ,'$'

; 1 2

; 012345678901234567890 1 2

myBoard db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

myBoard2 db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

myBoard3 db '+---+---+---+---+---+',0ah,0dh

db '| | | | | |',0ah,0dh

db '+---+---+---+---+---+',0ah,0dh,'$'

ends

stack segment

dw 128 dup(0)

ends

code segment

proc drawBoard ; draw Board .

push bp

mov bp,sp

push cx ; save before the change

lea dx, myBoard

mov ah, 09h

int 21h

pop cx ; Returns the value that was set before the change

pop bp ; returns bp before the changes

ret ; add ret beacuse i dont want again random numbers.

endp drawBoard ; end p. draw Board.

proc drawBoard2 ; draw Board .

push bp

mov bp,sp

push cx

lea dx, myBoard2

mov ah, 09h

int 21h

pop cx

pop bp

ret ; add ret beacuse i dont want again random numbers.

endp drawBoard2 ; end p. draw Board.

proc drawBoard3 ; draw Board .

push bp

mov bp,sp

push cx

lea dx, myBoard3

mov ah, 09h

int 21h

pop cx

pop bp

ret ; Returns to the place at the beginning of the procedure

endp drawBoard3 ; end p. draw Board.

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

lea dx, mygrapics ; this bring the grapics

mov ah, 09h

int 21h

; add your code here

; part one - bring the input from the user .

bringtheinput: ; here i will bring data from the user

lea dx,asktoinputdata

mov ah ,9

int 21h

; take the input and save in regi

; I want to input 5 numbers .

mov si,0

mov di,25 ; for the table start position.

mov cx,5

mov ax,0

call drawboard ; add numbers in new table

startloop2:

cmp cl,0

je randgen ; while mone!=5 beacuse I want 5 numbers

push cx

mov ah, 1

int 21h ;

mov arr2[si],al

mov [myBoard2 + di],al ; to add the input numbers to my table game. ; input saves in al .

inc si

add di, 4

call drawBoard2

loop startloop2

pop cx

RANDGEN: ; generate a rand no using the system time

mov cx,5

mov si,0 ; for the cells in the arr. to control in their location

startloop:

push cx

; part 2 - random numbers.

RANDSTART:

cmp cl,0

je continue ; while mone!=5 beacuse I want 5 random numbers

MOV AH, 00h ; interrupts to get system time

INT 1AH ; take number from the clock

MOV BH, 57 ; the max is 56 in ascii = 9

MOV AH, DL

CMP AH, BH ; compare with value in DL,

JA RANDSTART ; if more, regenerate. if not, continue...

MOV BH, 49 ; the min is 48 in ascii = 0

MOV AH, DL

CMP AH, BH ; compare with value in DL

JB RANDSTART ; if less, regenerate.

; if not:

; save the random in dl , copy dl to arr

mov arr[si],dl ; because the random numbers saves in dl.

mov ah, 5h ;5h output in new screen.

int 21h

inc si ; for the arrs.

pop cx

loop startloop

continue:

mov cx,5

mov si,0

mov di,25 ; start position .

lea dx,showrandomnumbersa

mov ah ,9

int 21h

showRandomNumbers: ; show the random numbers in new table .

mov dl,arr[si]

mov [myBoard + di], dl ;\*\*dl because there the random numbers,

add di, 4

inc si

call drawBoard ; draw a board for the random numbers. new table after add random numbers to the table.

loop showRandomNumbers

; part 3 - check if equel or not .

mov si,-1 ; make sure si = -1 for arr. -1 because i add +1 in the first,before the move.

mov cx,5 ; for the loop .

mov di,25 ; start beggining .

mov ax,0

comparing-arrays: ; comparing arrays

inc si ; check if this is the suitble place.

mov al,arr[si]

cmp arr2[si],al ; i cant compar memort to memory. i need oger maver in loop.

je equel ; if the numbers equel .

cmp arr2[si],al

jne theynotequel ; if they not equel.

equel: ; they equel

mov bx,0

mov bl,'T'

mov arrcheck[si],bl ;becuse i cant move from memory to memory.

jmp endLoop

theynotequel:

mov bx,0

mov bl,'F'

mov arrcheck[si], bl

endLoop:

loop comparing-arrays

; now i want to put T or F in another arr in board.

mov cx,5

mov di,25

mov si,0

mov dl,0

lea dx,checkyourguess

mov ah ,9

int 21h

; show T\F

startloop4: ;loop that put T or F in the board.

push cx ; check.........

mov dl, arrcheck [si]

mov [myBoard3 + di],dl ;to add the input F\T to my table game.

add di, 4

inc si ; for the arr.

call drawBoard3 ; to add t or f in the other board .

loop startloop4

pop cx ; bring cx before the changes .

lea dx, pkey

mov ah, 9

int 21h ; output string at ds:dx

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.