**מעבדה 10:**

מגישים:

איילון בן סימון – 312162951

סער ויקטור – 312392822

**קוד תרגיל 1:**

;lab10.asm

;int\*\* mallocAndInitMat(int n, int m, int (\*getVal)(int, int))

; BP+4 BP+6 BP+8

;int mulfunc(int i, int j)

.MODEL **SMALL**

.STACK 100h

.DATA

;defines variables

i DW 0

j DW 0

counter DW 0

.CODE

.386

;enabling calling the malloc function from c

EXTRN \_malloc **:** **NEAR**

\_mallocAndInitMat PROC **NEAR**

PUBLIC \_mallocAndInitMat

;save registers

**PUSH** **BP**

**MOV** **BP,SP**

**PUSH** **DI**

**PUSH** **SI**

**MOV** **DX,[BP+**4**]** ;DX=n

**MOV** counter**,DX** ;counter=n

**SHL** **DX,**1 ;DX=n\*2

**PUSH** **DX** ;PUSH n\*2 to stack

**CALL** \_malloc ;Calling the malloc function from c

;memory allocation check

**CMP** **AX,**0

**JE** sof ;if it doesn't work jump to sof

**ADD** **SP,**2

;save the address of the matrix in SI and DI

**MOV** **SI,AX**

**MOV** **DI,AX**

**MOV** **DX,[BP+**6**]** ;DX=m

**SHL** **DX,**1 ;DX=m\*2

;loop to allocate n columns, in size m

Memloop**:**

**CMP** counter**,**0

**JE** next

**PUSH** **DX**

**CALL** \_malloc ;Calling the malloc function from c

;memory allocation check

**CMP** **AX,**0

**JE** sof ;if it doesn't work jump to sof

**MOV** **[DI],AX**

**ADD** **SP,**2

**ADD** **DI,**2

**DEC** counter

**MOV** **DX,[BP+**6**]**

**SHL** **DX,**1

**JMP** Memloop

next**:**

**MOV** **DI,SI** ;DI will point to the start of the matrix again

**MOV** **DX,[BP+**6**]** ;DX=m

**MOV** **CX,[BP+**4**]** ;CX=n

;loop to switch to another row

L1**:**

**CMP** i**,CX** ;check if the matrix is full

**JE** sof

**MOV** **BX,[DI]**

**ADD** **DI,**2

**INC** i

**MOV** j**,**1

;loop to put value in every cell of the row

L2**:**

**CMP** j**,DX**

**JG** L1

**PUSH** i

**PUSH** j

**CALL** **[BP+**8**]** ;call getVal function from c

**ADD** **SP,**4

**MOV** **[BX],AX**

**ADD** **BX,**2

**INC** j

**MOV** **DX,[BP+**6**]**

**JMP** L2

sof**:**

**MOV** **AX,SI** ;return the poinetr of the matrix to c

**POP** **SI**

**POP** **DI**

**POP** **BP**

**RET**

\_mallocAndInitMat ENDP

END

**פלט תרגיל 1:**

