**תרגיל בית 3:**

מגישים:

איילון בן סימון – 312162951

סער ויקטור – 312392822

**קוד תרגיל 1:**

;targ3.asm

;getDiffMin(int \*mat[], int size , int \*num1 , int \*num2)

; BP+6 BP+10 BP+12 BP+16

.MODEL **LARGE**

.STACK 100h

.DATA

;define variables

res DW **?**

x DW **?**

y DW **?**

num DW 0

temp DW 0

nextrow DW 0

nextCell DW 0

i1 DW 0

i2 DW 0

i3 DW 0

i4 DW 0

tempAddr DW 0

off1 DW 0

off2 DW 0

seg1 DW 0

.CODE

.386

\_getDiffMin PROC **FAR**

PUBLIC \_getDiffMin

**PUSH** **BP** ;save the register BP

**MOV** **BP,SP** ;the BP register will point to the top of the stack

;save the registers in the stack

**PUSH** **SI**

**PUSH** **DI**

**PUSH** **ES**

**PUSH** **FS**

**PUSH** **GS**

**MOV** **SI,[BP+**6**]** ;SI=matrix OFF

**MOV** **ES,[BP+**8**]** ;ES=matrix SEG

**MOV** **CX,[BP+**10**]** ;pop the size of row in matrix

;initializing stopping condition for every loop

**MOV** i1**,CX**

**MOV** i2**,CX**

**MOV** i3**,CX**

**MOV** i4**,CX**

;initializing x,y and the first differnce (res)

**MOV** **BX,ES:[SI]**

**MOV** **FS,ES:[SI+**2**]**

**MOV** **AX,FS:[BX]**

**MOV** **DX,FS:[BX+**2**]**

**MOV** x**,AX**

**MOV** y**,DX**

**SUB** **AX,DX**

**JNS** positive

**NEG** **AX** ;converts the negative number to positive

positive**:**

**MOV** res**,AX**

;L1-loop to jump to the next row after checking the whole matrix

L1**:**

**CMP** i1**,**0

**JE** endlabel

**MOV** **DI,[BP+**6**]** ;DI=matrix OFF

**MOV** **ES,[BP+**8**]** ;ES=matrix SEG

**ADD** **DI,**nextrow ;jump to the next row with every itertion

;pop the array from the matrix (OFF and SEG)

**MOV** **BX,ES:[DI]**

**MOV** **FS,ES:[DI+**2**]**

;save the address of the row we are working on (SEG + OFF)

**MOV** tempAddr**,BX**

**MOV** off2**,FS**

**ADD** nextrow**,**4

**MOV** nextCell**,**0

**MOV** **CX,[BP+**10**]** ;pop the size of row in matrix

**MOV** i2**,CX**

**DEC** i1

;L2- loop to put the next value of the same row in AX

L2**:**

**CMP** i2**,**0

**JE** L1

;initializing the address of the row we are working on (SEG + OFF)

**MOV** **DI,**tempAddr

**MOV** **FS,**off2

**ADD** **DI,**nextCell

;save the address of AX to avoid cheking same cell in the matrix (SEG + OFF)

**MOV** off1**,DI**

**MOV** seg1**,FS**

**MOV** **AX,FS:[DI]**;save the value we are working on

**MOV** num**,**0

**ADD** nextCell**,**2

**MOV** **CX,[BP+**10**]** ;pop the size of row in matrix

**MOV** i3**,CX**

**DEC** i2

;L3- loop to jump to the next row

L3**:**

**CMP** i3**,**0

**JE** L2

**MOV** **SI,[BP+**6**]** ;SI=matrix OFF

**MOV** **ES,[BP+**8**]** ;ES=matrix SEG

**ADD** **SI,**num

**MOV** **BX,ES:[SI]**;BX=OFF

**MOV** **GS,ES:[SI+**2**]**;GS=SEG

**MOV** **DI,**0

**ADD** num**,**4

**MOV** **CX,[BP+**10**]** ;pop the size of row in matrix

**MOV** i4**,CX**

**DEC** i3

;lL4-loop to move on the same row

L4**:**

**CMP** i4**,**0

**JE** L3

**ADD** **BX,DI**

**MOV** **CX,GS**

;check if the difference is being made on the same cell

**CMP** seg1**,CX**

**JNE** pass

**CMP** off1**,BX**

**JE** next

pass**:**

**MOV** **DX,GS:[BX]**

**MOV** temp**,AX**

**SUB** **AX,DX**

**JNS** positive1

**NEG** **AX**

;check if AX<res update res,x and y

positive1**:**

**CMP** **AX,**res

**JGE** next

**MOV** res**,AX**

**MOV** **AX,**temp

**MOV** x**,AX**

**MOV** y**,DX**

next**:**

**MOV** **CX,**0

**MOV** **AX,**temp

**MOV** **DI,**2

**DEC** i4

**JMP** L4

;move the end values to c

endlabel**:**

**MOV** **BX,[BP+**12**]**

**MOV** **GS,[BP+**14**]**

**MOV** **AX,**x

**MOV** **GS:[BX],AX**

**MOV** **BX,[BP+**16**]**

**MOV** **GS,[BP+**18**]**

**MOV** **AX,**y

**MOV** **GS:[BX],AX**

**MOV** **AX,**res

;pop registers from the stack

**POP** **GS**

**POP** **FS**

**POP** **ES**

**POP** **DI**

**POP** **SI**

**POP** **BP**

;end of procedure

**RET**

\_getDiffMin ENDP

END

**פלט תרגיל 1:**

