# **M.Abdullah**

# **18F-0214**

# **Question 2:**

INCLUDE Irvine32.inc

.data

str1 BYTE "Rows : ",0

str2 BYTE "Columns : ",0

m1BYTE "1mat : ",0

m2 BYTE "2 mat : ", 0

m3 BYTE "result", 0

row1 BYTE ?

col1 BYTE ?

a1 BYTE 30 dup(? )

a2 BYTE 30 dup(? )

a3 BYTE 30 dup(?)

temp BYTE ?

row2 BYTE ?

col2 BYTE ?

x1 BYTE ?

sum BYTE ?

.code

main proc

mov edx, OFFSET str1

call writestring

call readdec

xor ecx,ecx

mov row1, al

mov edx,OFFSET str2

call writestring

call readdec

mov col1, al

mov cl, row1

mov esi, OFFSET a1

l1 :

push ecx

mov cl, col1

l2 :

call readdec

mov[esi], al

inc esi

loop l2

pop ecx

loop l1

mov cl, row1

mov esi, OFFSET a1

mov edx, OFFSET matx

call writestring

call crlf

l3 :

push ecx

mov cl, col1

l4 :

mov al, [esi]

call writedec

mov edx, OFFSET spac

call writestring

inc esi

loop l4

call crlf

pop ecx

loop l3

;2nd matrix

mov edx, OFFSET str1

call writestring

call readdec

xor ecx, ecx

mov row2, al

mov edx, OFFSET str1

call writestring

call readdec

mov col2, al

mov cl, row2

mov esi, OFFSET a2

mov cl, row2

mov esi, OFFSET a2

mov edx, OFFSET m2

call writestring

call crlf

l7 :

push ecx

mov cl, col2

l8 :

mov al, [esi]

call writedec

mov edx, OFFSET spac

call writestring

inc esi

loop l8

call crlf

pop ecx

loop l7

mov esi, offset a1

mov edx, offset a3

mov cl, row1

l9:

push ecx

mov x1,0

movzx ecx,col2

l10:

mov sum,0

mov ebx,OFFSET a2

mov al,x1

inc x1

add ebx,eax

push ecx

movzx ecx,row1

push esi

l11:

push edx

mov dl, [ebx]

mov al, [esi]

SUB dl

pop edx

add sum, al

inc esi

movzx eax, row2

add ebx, eax

loop l11

mov al, sum

mov[edx], al

inc edx

pop esi

pop ecx

loop l10

pop ecx

movzx eax, col1

add esi, eax

loop l9

mov edx, OFFSET m3

call writestring

xor ecx, ecx

mov cl, row2

mov esi, OFFSET a3

call crlf

l12 :

push ecx

mov cl, col2

l13 :

mov al, [esi]

call writedec

mov edx, OFFSET spac

call writestring

inc esi

loop l13

call crlf

pop ecx

loop l12

exit

main endp

end main

# **Question:2**

jc next

add ax,bx

jmp endfunction

next:

add ax,bx,1

end function: