# **18F-0214**

# **M.Abdullah**

# **Question:1**

INCLUDE Irvine32.inc

.data

A dword 10000 dup(0)

B dword 10000 dup(0)

num\_rows dword 0;

num\_cols dword 0;

col\_counter dword 0

row\_counter dword 0

counter byte 0

sum dword 0

.code

determinesize PROC

mov eax, matrix\_size

mov ebx, nextrow

mul ebx

mov num\_rows, eax

mov eax, matrix\_size

mov ebx, 4

mul ebx

mov num\_cols, eax

ret

determinesize ENDP

setvalue PROC

mov eax, upbound

call RandomRange

inc eax

mov [esi], eax

ret

setvalue ENDP

createA PROC

mov esi, OFFSET A

call traversematrix

ret

createA ENDP

createB PROC

mov esi, OFFSET B

call traversematrix

ret

createB ENDP

printvalue PROC

push eax

call WriteDec

call WriteChar

pop eax

ret

printvalue ENDP

traversematrix PROC

call Crlf

mov row\_counter, 0

Mov col\_counter, 0

Call setvalue

Call printvalue

Add esi, 4

Inc col\_counter

Cmp col\_counter, matrix\_size

Jle NewCol

Call Crlf

Inc row\_counter

Cmp row\_counter, matrix\_size

Jle NewRow

ret

addelements PROC

mov eax, working\_row

mov ebx, working\_col

mov row\_counter, eax

mov col\_counter, ebx

mov counter, 0

mov sum, 0

Call mul\_elements

Add sum, eax

Add row\_counter, 4

Add col\_counter, nextrow

Inc counter

Cmp counter, matrix\_size

Jle Next

Mov eax, sum

ret

addelements ENDP

mul\_elements PROC

mov esi, OFFSET A

add esi, row\_counter

mov eax, [esi]

mov esi, OFFSET B

add esi, col\_counter

mov ebx, [esi]

mul ebx

ret

mul\_elements ENDP

multiplymatrix PROC

call Crlf

mov working\_row, 0

Mov working\_col, 0

Call addelements

Call printvalue

Add working\_col, 4

Mov eax, working\_col

Cmp eax, num\_cols

Jle NewCol

Call Crlf

Add working\_row, nextrow

Mov eax, working\_row

Cmp eax, num\_rows

Jle NewRow

Call Crlf

ret

multiplymatrix ENDP

main proc

call Randomize

mov ebx, num\_matrices

INVOKE TimerStart,

ADDR timer1

Call createA

Call createB

Call determinesize

Call multiplymatrix

Inc newmatrix

Mov eax, newmatrix

Cmp eax, num\_matrices

Jle NextMatrix

ADDR timer1

Call WriteDec

Mov edx, OFFSET msg

Call WriteString

exit

main endp

end main