*Program 14:*

INCLUDE Irvine32.inc

.data

A dword 10000 dup(0)

B dword 10000 dup(0)

num\_rows dword 0;

num\_colsdword 0;

col\_counterdword 0

row\_counterdword 0

counterbyte 0

sumdword 0

.code

determinesize PROC

moveax, matrix\_size

movebx, nextrow

mulebx

movnum\_rows, ea*x*

moveax, matrix\_size

movebx, 4

mulebx

movnum\_cols, eax

ret

determinesize ENDP

setvalue PROC

moveax, upbound

callRandomRange

inceax

mov[esi], eax

ret

setvalue ENDP

createA PROC

movesi, OFFSET A

calltraversematrix

ret

createA ENDP

createB PROC

movesi, OFFSET B

calltraversematrix

ret

createB ENDP

printvalue PROC

pusheax

callWriteDec

moval, ' '

callWriteChar

popeax

ret

printvalue ENDP

traversematrix PROC

callCrlf

movrow\_counter, 0

NewRow:

*M*ovcol\_counter, 0

NewCol:

*C*allsetvalue

callprintvalue

addesi, 4

inccol\_counter

cmpcol\_counter, matrix\_size

jleNewCol;end inner loop

*C*allCrlf

incrow\_counte*r*

cmprow\_counter, matrix\_size

jleNewRow

ret

addelements PROC

moveax, working\_row

movebx, working\_col

movrow\_counter, eax

movcol\_counter, ebx

movcounter, 0

movsum, 0

Next*:*

*C*al*l* mul\_elements

addsum, eax

addrow\_counter, 4

addcol\_counter, nextrow

inccounte*r*

cmpcounter, matrix\_size

jleNe*xt*

moveax, sum

ret

addelements ENDP

mul\_elements PROC

movesi, OFFSET A

addesi, row\_counter

moveax, [esi]

movesi, OFFSET B

addesi, col\_counter

movebx, [esi]

mulebx

ret

mul\_elements ENDP

multiplymatrix PROC

cal*l* Crlf

movworking\_row, 0

NewRow:

*M*ovworking\_col, 0

NewCol:

*C*alladdelements

callprintvalue

addworking\_col, 4

moveax, working\_col

cmpeax, num\_cols

jleNewCol

callCrlf

addworking\_row, nextrow

moveax, working\_row

cmpeax, num\_rows

jleNewRow

callCrlf

ret

multiplymatrix ENDP

main proc

callRandomize

movebx, num\_matrices

INVOKETimerStart,

ADDR timer1

NextMatrix:

*C*allcreateA

callcreateB

calldeterminesize

callmultiplymatrix

incnewmatrix

moveax, newmatrix

cmpeax, num\_matrices

jleNextMatrix

INVOKETimerStop,

ADDR timer1

*C*allWriteDec

movedx, OFFSET msg

call WriteString

exit

main endp

end main