

Name: KEY

Id#

COE 205, Term 061
Computer Organization & Assembly Programming

Quiz# 3

Date: Sunday, Nov. 19, 2006

Q1. You are required to write a procedure, **ADISPLAY**, to display a two-dimensional array of unsigned integers. The procedure receives the address of the array in register EDX, the number of rows in register ESI and the number of columns in register EDI. The procedure has to preserve all the used registers. Then, use the procedure **ADISPLAY** to display the array **TARRAY** shown below:

```
TARRAY    DD 10, 20, 300, 400, 5000
           DD 60, 70, 1000, 2000, 2
           DD 99, 16, 11000, 50, 30
```

Use the procedure **WriteDec** for displaying the unsigned integers. This procedure displays the content of the register in EAX. Note that you need to display a space between integers displayed in the same row. You can use the procedure **WriteChar** to display the character stored in register AL. You need also to display each row in a new line. You can use the procedure **Crlf** for displaying a new line.

```
.686
.MODEL FLAT, STDCALL
.STACK

INCLUDE Irvine32.inc

.DATA

; (insert variables here)
TARRAY    DD 10, 20, 300, 400, 5000
           DD 60, 70, 1000, 2000, 2
           DD 99, 16, 11000, 50, 30

.CODE
main PROC

; (insert executable instructions here)
MOV EDX, offset TARRAY
MOV ESI, 3
MOV EDI, 5
CALL ADISPLAY

    exit        ; exit to operating system
main ENDP
```

```
; (insert additional procedures here)
```

```
-----  
; ADISPLAY: Displays a two-dimensional array of unsigned integers  
; Receives: EDX contains the address of the array, ESI contains the  
; number of rows, EDI contains the number of columns.  
; Returns: nothing  
; Requires: nothing  
-----
```

```
ADISPLAY PROC  
; saving used registers  
PUSH EDX  
PUSH ECX  
PUSH EAX  
  
MOV ECX, ESI; number of rows  
Nextr:  
PUSH ECX  
  
; displaying next row  
MOV ECX, EDI; number of columns  
Nextc:  
  
; displaying next column  
MOV EAX, [EDX]  
CALL WriteDec  
ADD EDX, 4  
  
; displaying a space between integers  
MOV AL, ' '  
Call WriteChar  
LOOP Nextc  
  
; displaying a new line between rows  
CALL Crlf  
POP ECX  
LOOP Nextr  
  
; restoring used registers  
POP EAX  
POP ECX  
POP EDX  
  
RET  
ADISPLAY ENDP
```

```
END main
```