Practice Assembly program before Midterm Exam

Dept. of CSIE, Fu Jen Catholic University
Hsin Chuang, 24205, Taipei Taiwan.
周賜福

Fourth program Question is as follows.

Calculate the Factorial numbers between N1 and N2:

Write a program to write a factorial between N1 to N2 where the N1 and N2 are entered by the user. The program that uses a loop to calculate **the first N2- N1 values** of the Factorial number sequence, described by the following formula: Fact(1)=1 Fact(2)=1
Fact(3)=1*2*3=6
Store the result in the facarray and display the result.

N	1	2	3	4	5	6	7	8	9	10	11	
Fact(n)	1	1	6	24	120	720	5040	40320	362880	3628800		

```
.data
factarray dword  20 DUP(0)
count EQU (LENGTHOF dwarray)
```

```
; Chapter 4 Exercise 2 Given on Nov 2nd, 2017 Demonstration of writing in an array
; This program reads Nimul1 and N2, calculate the factorial of N1 to N2.
; This program uses to multiply each element for the factorial.
;-----------------------------------------------------------------

INCLUDE Irvine32.inc
TITLE Asm(first.asm)

.data
ODDARRAY DWORD 200 DUP(0)
Message1 BYTE "Please Enter an integer N1 and then N2: Now enter N1", 0dh, 0ah, 0
Message2 BYTE "Please Enter an integer N2", 0dh, 0ah, 0
Message3 BYTE  "The total amount of factorial between N1 to N2 that you input is as follows: ", 0dh, 0ah,
0
N1 DWORD ?
N2 DWORD ?
Total DWORD 1
```

```
.code
main PROC
mov edx, OFFSET Message1          ; Move the first message into the register
call writeString                   ; Write Message1 on to the screen
call ReadInt                       ; Read the first input n1 from the user
call Crlf                          ; display Newline
mov  N1, eax                       ; Move the value entered into n1
        ; Now here we try to read N2 using the message 2 and reading in from the user.
mov edx, OFFSET Message2          ; Move the first message into the register
call writeString                   ; Write Message1 on to the screen
call ReadInt                      ; Read the second input n2 from the user
call Crlf                         ; display Newline
mov  N2, eax                       ; Move the value entered into n2

         ; Now we calculate the N2 - N1 to run the loop.
mov eax, N2                        ; move n2 into eax
sub eax, N1                        ; subtract n1 to see how many to calculate
mov ecx, eax                       ; move n to ecx
inc ecx                           ; you need to have one extra
mov esi, 0                        ; first pointer to array
mov eax, N1                       ;  move  N1 into eax
```

```
; The following loop writes in an array the value.


L1:
mov ODDARRAY[esi], eax          ;  move each value into the array
inc eax                             ; increment N1 to the next element.
add esi, TYPE DWORD             ; increment the pointer
loop L1                             ; Loop next element

        ; I adjust the loop to run the second time.
mov eax, N2                     ; Move N2 into eax
sub eax, N1                     ; subtract the N1
mov ecx, eax                    ; Move eax into ecx
inc ecx                         ; Increment one element.
mov esi, 0                      ; move the pointer zero
mov eax, 1                      ; move just one into eax.
```

```
        ; The following loop reads from an array one by one.
        ; Then multiplies with the next element to create factorial for the  total

L2 :
mov ebx, ODDARRAY[esi]          ; Move the first element into eax
imul eax, ebx                    ; Multiply with the next element.
call writedec                    ; Print that element to the screen.
call crlf                        ; Move next line.
add esi, TYPE DWORD              ; Add pointer to esi
loop L2                          ; Loop again.

mov Total, eax
                          ;Get ready to print the result
mov edx, OFFSET Message3         ; Move the string offset
call writestring                 ; Write that string.
call crlf                        ;  Move the next line character.
mov eax, Total                   ; Move the total into eax
call writedec                    ; Write the total to the screen
call crlf                        ; Next line.
exit                             ; Exit gracefully.
main endp
end main
```

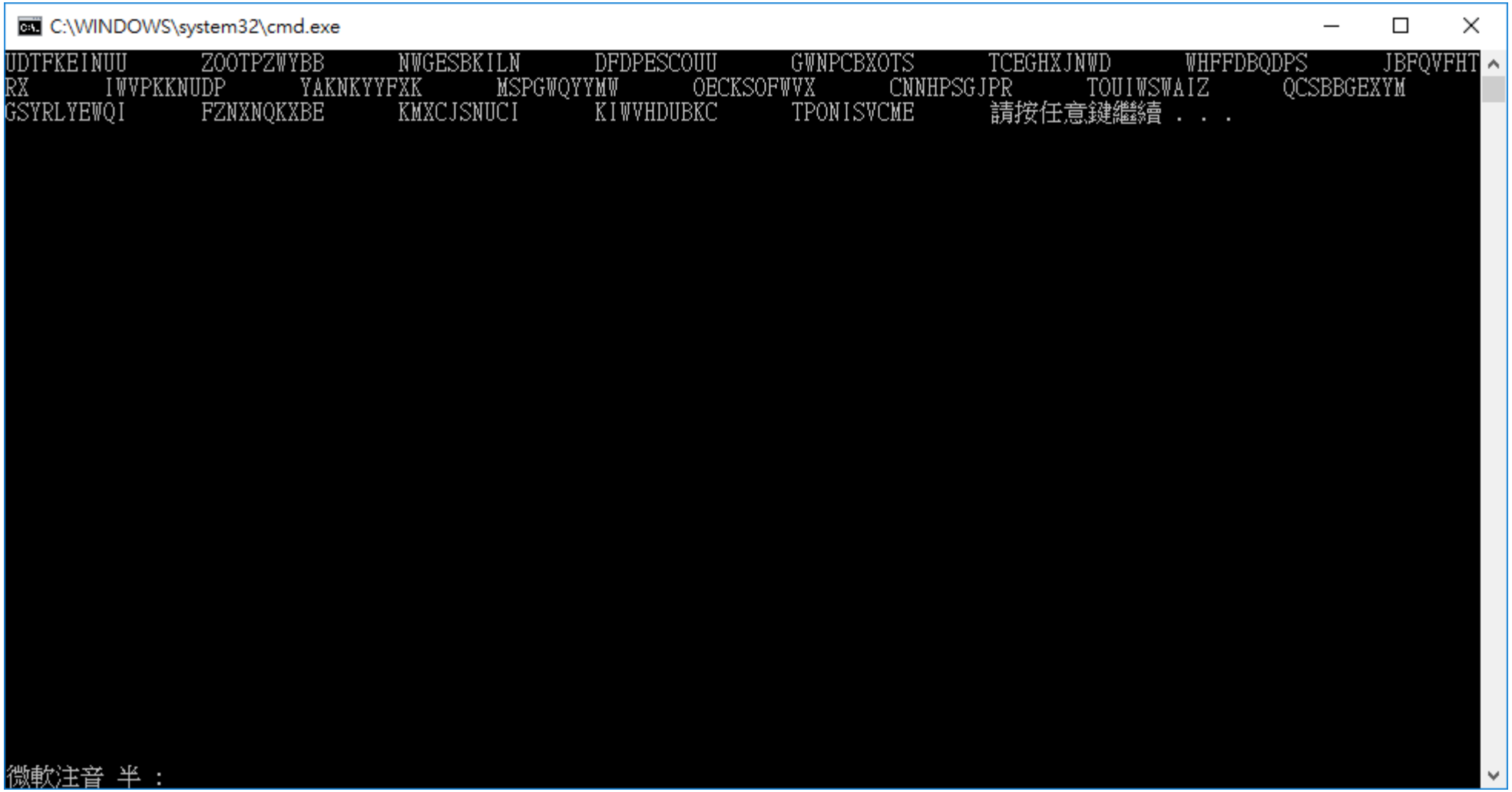
Second program Question is as follows.

Random Strings:

Create a procedure that generates a random string of length L, containing all capital letters. When calling the procedure, pass the value of L in EAX, and pass a pointer to an array of byte that will hold the random string.

Write a test program that calls your procedure 20 times and displays the strings in the console window.

SAMPLE output from the program



Fifth program Question is as follows.

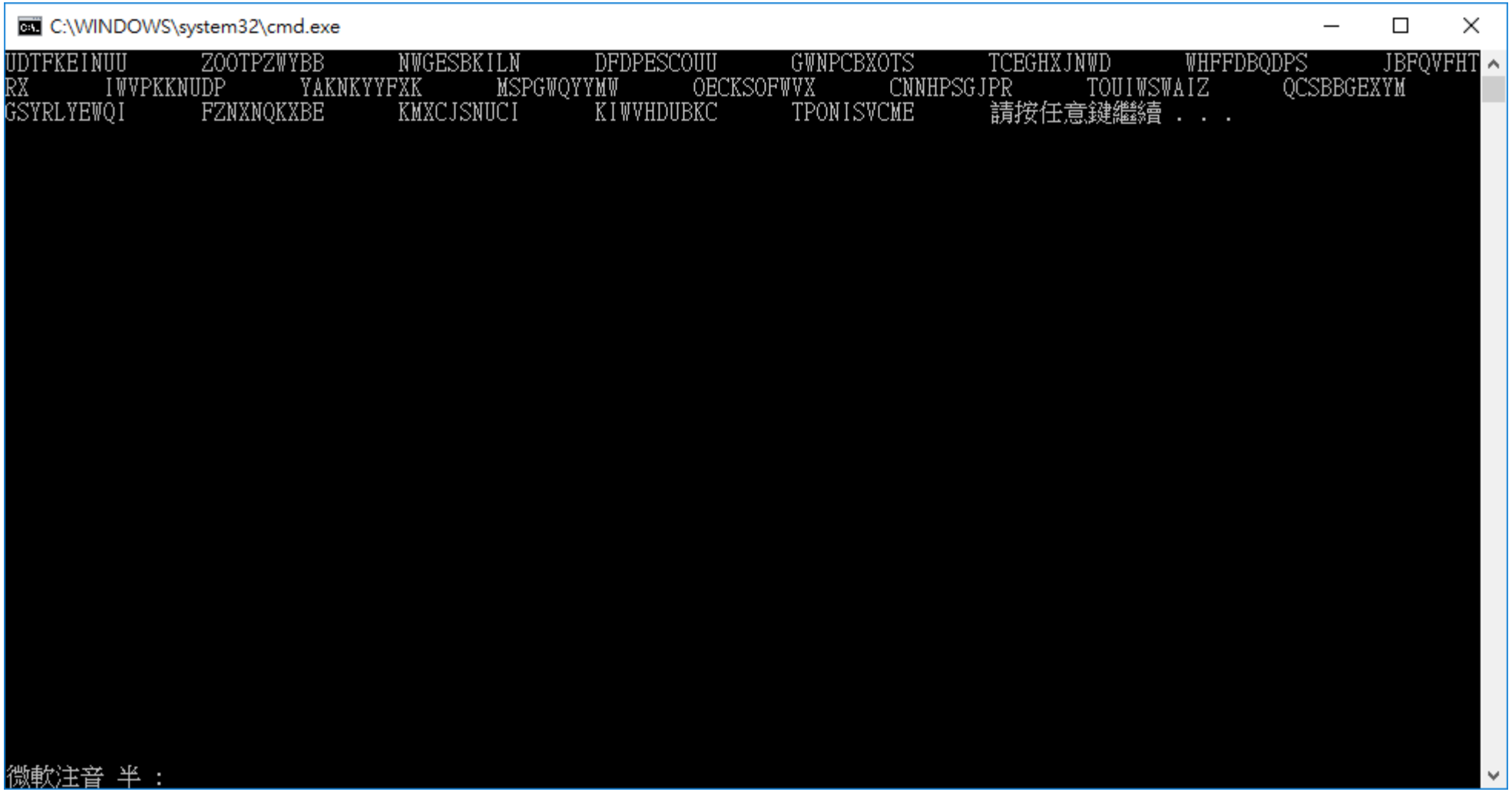
; Exercise 6: Random Strings
Comment !
Description : Create a procedure that generates a random string of length L, containing all capital letters. When calling the procedure, pass the value of L in EAX, and pass a pointer to an array of byte that will hold the random string.Write a test program that calls your procedure 20 times and displays the strings in the console window
!

```
INCLUDE Irvine32.inc

STR_COUNT = 20
STR_SIZE = 10
TAB = 9
.data
aString BYTE STR_SIZE DUP(0), 0
```

```
.code
main PROC
mov  ecx, STR_COUNT       ; loop count
call randomize
L1 :
; generate a single string
mov   eax, STR_SIZE; loop counter
mov   esi, OFFSET aString          ; string index
call  CreateRandomString

mov  edx, OFFSET aString          ; display the string
Call  WriteString
mov al, TAB                       ; after each string place a tab
call writechar
                                  ; callCrlf
loop  L1
exit
main ENDP
```

```
; ---------------------------------------- -
CreateRandomString PROC
; Creates a random string.  Receives: ESI points to the string,
; EAX contains the string size.  Returns:  nothing
; ---------------------------------------- -
push   ecx
mov   ecx, eax              ; get string size
L1 :
mov   eax, 26               ; generate random int(0..25)
call  RandomRange
add  eax, 'A‘               ; range: 'A'..'Z'
mov [esi], al               ; store the character
inc esi                     ; next character position
Loop L1                     ; Loop here.
pop  ecx
ret
CreateRandomString ENDP
END main
```

SAMPLE output from the program



END of our presentation