Practice Program exercises Assembly program組合語言實驗-資訊工程學系 2年級

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

周賜福-Oct19th, 2018

First program Question is as follows.

Program to exchange pairs of Array values: **page 169**

Write a program with a loop and indexed addressing that exchanges every pair of values in an array with an even number of elements. Therefore item i will exchange with item i+1, and item i+2 with item i+3, and so on. Assume we have only even number of elements in an array.

[Given the following array with 10 elements. Just use a DWORD array if you wish:]

Ex1: {21, 32, 45, 64, 76, 87, 56, 23, 34, 77}

Result of running your program would be { 32, 21, 64, 45 87, 76, 23, 56, 77, 34}

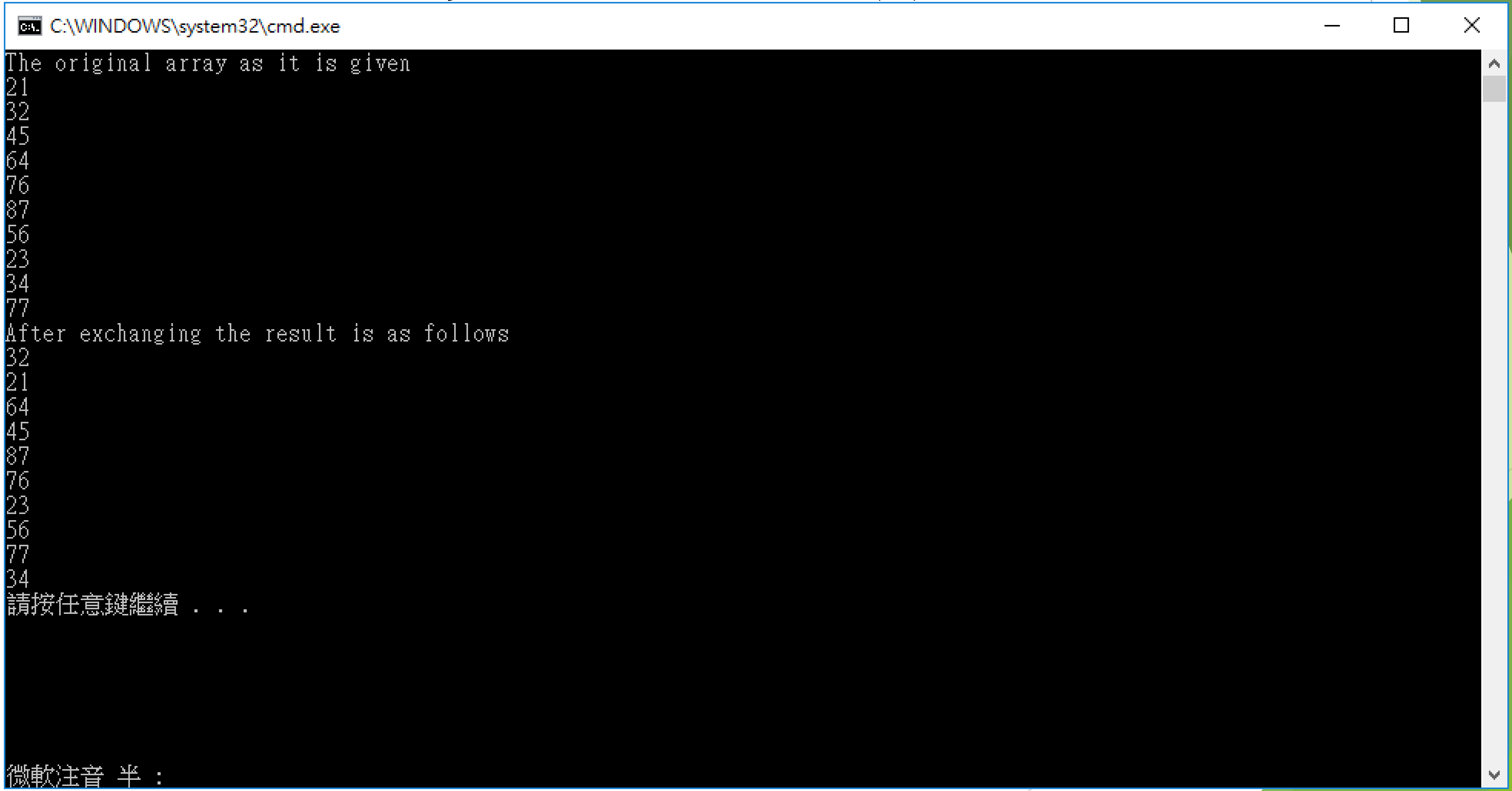
Result **must be stored in another array** which has empty elements.

Ex2: {0,2,5,9,10,12} Result of running your program {2,0,9,5,12,10}

```
.data
firstarray DWORD 21, 32, 45, 64, 76, 87, 56, 23, 34, 77
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```

Result of Running your program would look like this.

```
firstarray DWORD 21, 32, 45, 64, 76, 87, 56, 23, 34, 77
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```



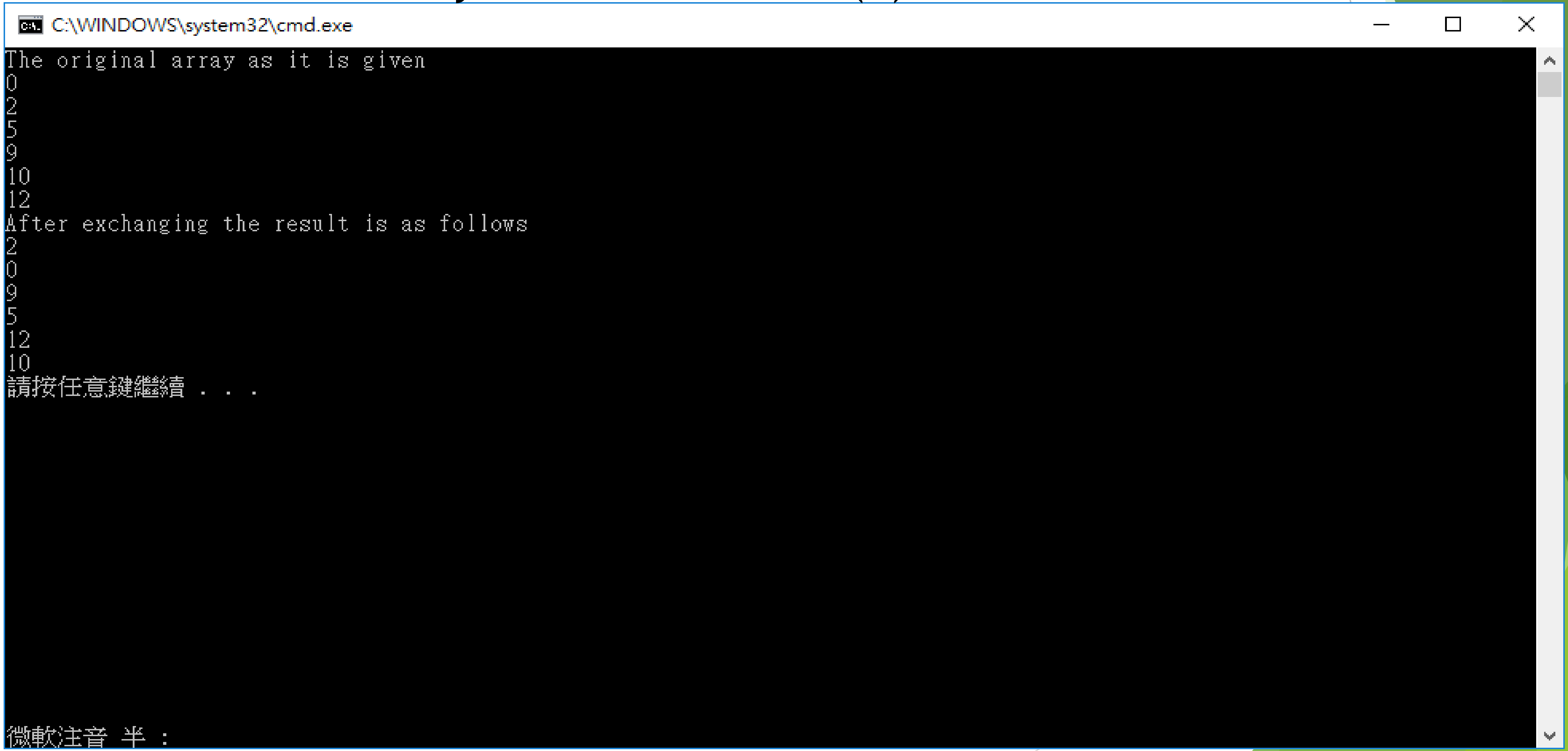
```
The original array as it is given
21
32
45
64
76
87
56
23
34
77
After exchanging the result is as follows
32
21
64
45
87
76
23
56
77
34
請按任意鍵繼續 . . .




微軟注音 半 :
```

Result of running example with the different element in the array

```
firstarray DWORD 0,2,5,9,10,12
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```



```
The original array as it is given
0
2
5
9
10
12
After exchanging the result is as follows
2
0
9
5
12
10
請按任意鍵繼續 . . .




微軟注音 半 :
```

Program to exchange pairs of Array values: page 169

Page 1/5

```
; Author: 周賜福
; Date Written: Oct 18th, 2018
;  Exercise 2: This program Exchanges Pairs of Array Values
; Here is an example of array 21, 32, 45, 64, 76, 87, 56, 23, 34, 77
; After exchanging of item i with i+1 and item i+2 with i+3 and so on.
; Result will be as follows:
; 32, 21, 64, 45 87, 76, 23, 56, 77, 34
; Date Written:   Oct 18th, 2018

TITLE ASM                    (oddprog.asm)
include Irvine32.inc
```

```
.data
str1 BYTE "The original array as it is given",0dh,0ah,0
str2 BYTE "After exchanging the result is as follows",0dh,0ah,0
dwarray dword  21, 32, 45, 64, 76, 87, 56, 23, 34, 77
count EQU (LENGTHOF dwarray)
                    ; result array as much as the dwarray
rearray dword count DUP(0)
```

Page 3/5

```
.code
main proc
    mov   ecx,count
    mov esi,0
L1:
    mov  eax,  dwarray[esi]       ; move the first element in to eax
    mov  rearray[esi+4], eax      ; move the element to the second place in result array
    mov  eax, dwarray[esi+4]      ; move the second element into eax
    mov rearray[esi], eax         ; move the eax into the first place into the result array
    add  esi,8                    ; increment two element into the esi to proceed ahead.
    loop L1                       ; call the loop here.
```

Page 4/5

```
    mov edx, OFFSET str1          ; move to print the source array
    call writestring               ; print the string
     mov ecx, count               ; move the counter to loop again
    mov esi, OFFSET dwarray   ; move the source array pointer to esi

L2:
   mov eax, [esi]             ; move the first element into the register.
    call writedec             ; print the string
    call crlf                  ; call next line
    add esi, TYPE dwarray       ; increment the pointer.
    loop L2
```

Page 5/5

```
     mov edx, OFFSET str2           ; move to print the result array
     call writestring               ; print the string
     mov ecx, count                ; get ready for another loop here
     mov esi, OFFSET rearray        ; move offset of result array to print
                  ; Get ready to loop again to print the result.
   L3:                              ; Label L3 different from L2 label.
    mov eax, [esi]                 ; using the pointer to array move the first element
    call writedec                  ; Write the value you from the result array
    call crlf                      ; change the line.
    add esi, TYPE rearray          ; increment the pointer to print next element.
    loop L3                         ; Loop here.

     invoke ExitProcess,0          ; exit gracefully.
  main endp
  end main
```

Another program given in the following page

Second program Question is as follows.

Shifting the array elements in an Array   page 170

Using a loop and indexed addressing, write code that rotates the members of a 32 bit integer array forward one position. The value at the end of the array must wrap around to the first position. For example, Look at the scenario below:

[Given the following array with 13 elements. Just use a DWORD array if you wish:]

Ex1:   {21, 32, 45, 64, 76, 87, 56, 23, 34, 77,83,99,103} [Last element 103 is written first and the rest of the elements are moved backward.]

Result of running your program would be {103, 21, 32, 45, 64, 76, 87, 56, 23,34,77,83,99}

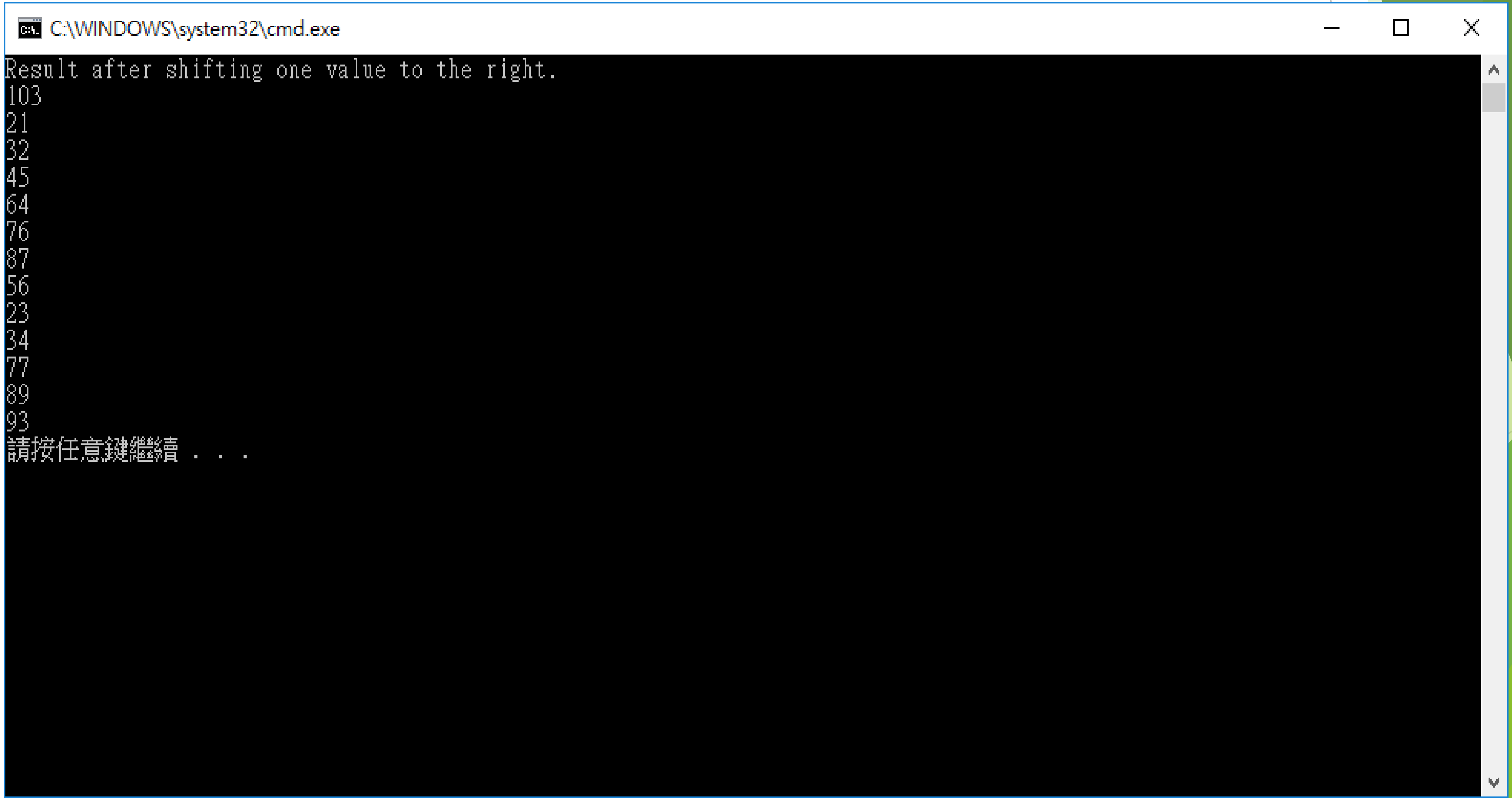
Result **must be stored in another array** which has empty elements.

Ex2: 9 elements in the array   {0,2,5,9,10,12,19,23,34}   Result of running your program {34,0,2,5,9,10,12,19,23}

```
.data
firstarray DWORD 21, 32, 45, 64, 76, 87, 56, 23, 34, 77,83,99,103
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```

Result of Running your program would look like this.

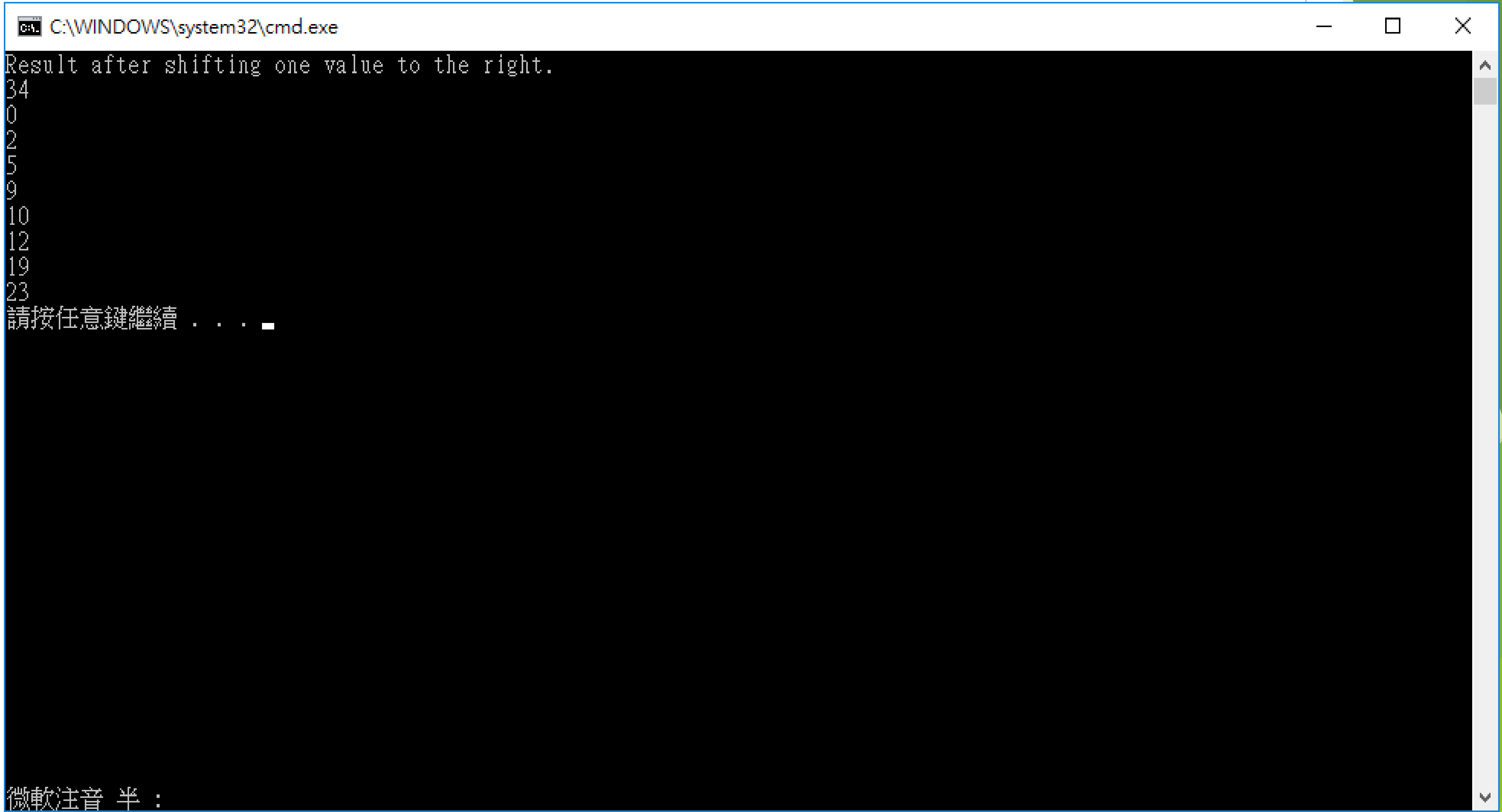
```
firstarray DWORD 21, 32, 45, 64, 76, 87, 56, 23, 34, 77,83,99,103
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```



```
Result after shifting one value to the right.
103
21
32
45
64
76
87
56
23
34
77
89
93
請按任意鍵繼續 . . .
```

Result of running example with the different element in the array

```
firstarray DWORD 0,2,5,9,10,12,19,23,34
count  EQU (LENGTHOF firstarray)
resultarray dword count DUP(0)
```



```
Result after shifting one value to the right.
34
0
2
5
9
10
12
19
23
請按任意鍵繼續 . . .
```

Shifting the array elements in an Array- Assembly program

Page 1/6

```
TITLE Inclass exercise No2.   inclass2.asm
INCLUDE Irvine32.inc
ExitProcess proto,dwExitCode:dword   (optional)
 ; Shifting an Array
 ; This program shifte one value to the next position and the last value is written
 ; to the first element of the array.

Comment !
    Using a loop and indexed addressing, write code that rotates the members
    of a 32-bit integer array forward two position. The value at the end of
    the array must wrap around to the first position. For example, the array
    [21, 32, 45, 64, 76, 87, 56, 23, 34, 77, 89, 93, 103] would be transformed int
    [103, 21, 32, 45, 64, 76, 87, 56, 23, 34, 77, 89, 93].
    Another array to use is as follows [0,2,5,9,10,12,19,23,34]
!
```

Shifting the array elements in an Array- Assembly program

Page 2/6

```
    ; Declare the following Data section here below.
.data
array dword 21, 32, 45, 64, 76, 87, 56, 23, 34, 77, 89, 93, 103
count EQU (LENGTHOF array)

resultarray DWORD count DUP(0)

MSG0 BYTE  "Result after shifting one value to the right.",0dh,0ah,0
```

Shifting the array elements in an Array- Assembly program

Page 3/6

```
.code
main proc
   mov esi, OFFSET array              ; store the pointer to the array here
    mov edi, OFFSET resultarray       ; store the pointer to the result array.
    mov  eax, array[4 * (count-1)]; save last value of the array here
    mov  ecx, count-1                  ; Run the loop one less.
```

Shifting the array elements in an Array- Assembly program

Page 4/6

```
L1:
   mov edx, [esi]          ; move the first element of the array in edx,
   mov [edi], eax          ; move last element of the array in the first
                           ;    element
   mov eax, edx            ; exchange these two, next time first element
                           ; will be written in the next.
  add  esi, TYPE array        ; increment the pointer of the array element.
   add  edi, TYPE resultarray      ; increment the pointer of the
                                     ; result array element.
   loop L1                    ; Now loop
   mov [edi],eax       ; Here I write the n-1 element in the Nth position.
```

Shifting the array elements in an Array- Assembly program

Page 5/6

```
 ; I get ready to print the result and move the string of result
    mov edx, OFFSET MSG0          ; move the pointer to the string in edx
    call writestring               ; print the string to say, I have the resul
    mov ecx, count                ; move the count to loop N times
    mov esi, OFFSET resultarray     ; move the offset of the result array.
              ; Here I begin to loop to print the result
L2:
        mov eax, [esi]              ; using the pointer, move the first element
     call writedec                 ; write the element here.
     call crlf                      ; next line of the value
     add esi, TYPE resultarray       ; increment the pointer to the next element.
     loop L2                      ; Here I loop.
```

Shifting the array elements in an Array- Assembly program

Page 6/6

```
exit                                    ; finish here gracefully and so exit.
main endp
end main
```

Finish the Second program Exercise.

Let us take the Third program and see how we can write the program.

Third program Question is as follows.

Summing the Gaps between Array values **page 169**

Write a program with a loop and indexed addressing that calculates the sum of all the gaps between successive array elements. The array elements are doublewords, sequenced in non-decreasing order. So, for example the array {0,2,5,9,10} has gaps of 2, 3, 4, and 1, whose sum equals 10.

Hint: The gap between 0 and 2 is = 2, gap between 2 and 5 is =3, gap between 5 and 9 = 4, gap between 9 and 10 is =1. (Result of adding 2 +3+4+1= 10.

Another example {12, 15, 22, 29, 37, 43} Result is = 3+7+7+8+6= 31

```
.data
dwarray dword  12, 15, 22, 29, 37, 43
count EQU (LENGTHOF dwarray)
```