Microsoft (R) Macro Assembler Version 12.00.31101.0 03/20/15 02:20:30

example.asm Page 1 - 1

; Author: Brandon Lundberg

; File Name: lab7.asm

; Purpose: Selection sort

; Date: 19 March 2015

; Input: A string of 10 numbers

; Output: Output of the original array, and then the sorted array in one output box

.586

.MODEL FLAT

INCLUDE io.h

C ; IO.H -- header file for I/O macros (listing suppressed)

C .NOLIST ; turn off listing

C .LIST ; begin listing

C

.STACK 4096

00000000 .DATA

00000000 45 6E 74 65 72 prompt1 BYTE "Enter 10 scores separated by a space",0

20 31 30 20 73

63 6F 72 65 73

20 73 65 70 61

72 61 74 65 64

20 62 79 20 61

20 73 70 61 63

65 00

00000025 6F 75 74 70 75 prompt2 BYTE "output is: ",0

74 20 69 73 3A

20 00

00000031 00000050 [ scoreString BYTE 80 dup(?),0; //for 10 scores - make enough room

00

] 00

00000082 0000000A [ scoreArray DWORD 10 dup (?)

00000000

]

000000AA 0000000B [ temp BYTE 11 dup(?),0

00

] 00

000000B6 0000000B [ tempx BYTE 11 dup(" "),0; //or,..dup(20H); 20H=space

20

] 00

000000C2 0000000B [ temp2 BYTE 11 dup(?),0

00

] 00

000000CE 4F 72 69 67 69 outputstring BYTE "Original: ", 250 DUP(20H), 0 ; Size 10

6E 61 6C 3A 20

000000FA [

20

] 00

000001D3 53 6F 72 74 65 label1 BYTE "Sorted: ",0

64 3A 20 00

000001DC 00000000 count1 DWORD 0

000001E0 00000000 count2 DWORD 0

000001E4 0000000A [ sorted DWORD 10 dup (?)

00000000

]

00000000 .CODE

00000000 \_MainProc PROC

input prompt1, scoreString, 40; //gets 10 scores

0000001E 8D 1D 00000031 R lea ebx, scoreString

00000024 FF 05 000001DC R outerLoop: inc count1; //outer loop counter++

0000002A 8D 35 000000B6 R lea esi, tempx; //flush temp string before using

00000030 8D 3D 000000AA R lea edi, temp

00000036 FC cld

00000037 B9 0000000B mov ecx, 11

0000003C F3/ A4 rep movsb

0000003E 8D 15 000000AA R lea edx, temp;

00000044 80 3B 20 innerLoop: cmp byte ptr[ebx], 20h; //if ending mark(space), done

00000047 74 0D je done1;

00000049 80 3B 00 cmp byte ptr[ebx], 00h; //elsif null char, also done

0000004C 74 08 je done1;

0000004E 8A 03 mov AL, byte ptr[ebx]; //otherwise, get 1 byte from input string

00000050 88 02 mov [edx], AL; //and move it to temp

00000052 43 inc ebx; //to next byte in input string

00000053 42 inc edx; //to next byte in temp string

00000054 EB EE jmp innerLoop; //inner loop (temp <- one score)

00000056 done1: output prompt2, temp; test display of temp

atod temp; //eax <- temp

0000007E 8B 0D 000001DC R mov ecx, count1;

00000084 49 dec ecx; //counter:1 -> array index:0

00000085 6B C9 04 imul ecx, 4; //array ele size = 4 bytes

00000088 89 81 00000082 R mov scoreArray[ecx], eax; //store one score in array

0000008E 43 inc ebx; //skip the end mark(space) in the input string

0000008F 83 3D 000001DC R cmp count1, 10; //loop 10 times

0A

00000096 7C 8C jnge outerLoop

dtoa temp2, scoreArray[0]; or, scorearray+0; //testing display

output prompt2, temp2

; Set array counter to zero

000000CD B9 00000000 mov ecx,0

; Copy data to sorted array

000000D2 Copy:

000000D2 8B 81 00000082 R mov eax, scoreArray[ecx]

000000D8 89 81 000001E4 R mov sorted[ecx], eax

000000DE 83 C1 04 add ecx, 4 ; Increase counter to next value

000000E1 FF 05 000001E0 R inc count2

000000E7 83 3D 000001E0 R cmp count2, 10

0A

000000EE 7C E2 jl Copy

; Selection sort

000000F0 BB 00000000 mov ebx, 0 ; First index

000000F5 B9 00000000 mov ecx, 0 ; First index

000000FA BA 00000004 mov edx, 4 ; Second index

000000FF OutLoop:

000000FF 8B 83 000001E4 R mov eax, sorted[ebx] ; Get value of current element

00000105 InLoop:

; Find index for current smallest element

00000105 3B 82 000001E4 R cmp eax, sorted[edx]

0000010B 7F 2B jg MinValue

0000010D Continue:

; Check counter for inner loop

0000010D 83 C2 04 add edx, 4

00000110 83 FA 28 cmp edx, 40

00000113 7C F0 jl InLoop

; Swap current index with highest value

00000115 8B 81 000001E4 R mov eax, sorted[ecx]

0000011B 87 83 000001E4 R xchg sorted[ebx], eax

00000121 89 81 000001E4 R mov sorted[ecx], eax

; Check counter for outer loop

00000127 83 C3 04 add ebx, 4

0000012A 8B CB mov ecx, ebx

0000012C 8B D3 mov edx, ebx

0000012E 83 C2 04 add edx, 4

00000131 83 FB 24 cmp ebx, 36

00000134 7C C9 jl OutLoop

; Jump to printing

00000136 EB 0A jmp PrintResults

00000138 MinValue:

00000138 8B CA mov ecx, edx

0000013A 8B 82 000001E4 R mov eax, sorted[edx]

00000140 EB CB jmp Continue

00000142 PrintResults:

; Append scoreArray to output string

00000142 BA 00000000 mov edx, 0

00000147 BB 0000000A mov ebx, 10

0000014C OutputOriginal:

; Print the results of the first array

0000014C 8B 82 00000082 R mov eax, scoreArray[edx]

dtoa temp2, eax

0000016A 8D 35 000000C3 R lea esi, temp2[1]

00000170 8D BB 000000CE R lea edi, outputstring[ebx]

00000176 FC cld

00000177 B9 0000000A mov ecx, 10

0000017C F3/ A4 rep movsb

0000017E 83 C3 0A add ebx, 10

00000181 83 C2 04 add edx, 4

00000184 83 FA 28 cmp edx, 40

00000187 7C C3 jl OutputOriginal

00000189 C6 05 0000013C R mov outputstring + 110, 0AH

0A

; Sorted label to output

00000190 8D 35 000001D3 R lea esi, label1

00000196 8D 3D 0000013D R lea edi, outputstring + 111

0000019C FC cld

0000019D B9 00000008 mov ecx, 8

000001A2 F3/ A4 rep movsb

000001A4 C6 05 00000145 R mov outputstring + 119, 20H

20

000001AB C6 05 00000146 R mov outputstring + 120, 20H

20

000001B2 C6 05 00000147 R mov outputstring + 121, 20H

20

000001B9 BA 00000000 mov edx, 0

000001BE BB 0000007A mov ebx, 122

000001C3 OutputSorted:

000001C3 8B 82 000001E4 R mov eax, sorted[edx]

dtoa temp2, eax

000001E1 8D 35 000000C3 R lea esi, temp2[1]

000001E7 8D BB 000000CE R lea edi, outputstring[ebx]

000001ED FC cld

000001EE B9 0000000A mov ecx, 10

000001F3 F3/ A4 rep movsb

000001F5 83 C3 0A add ebx, 10

000001F8 83 C2 04 add edx, 4

000001FB 83 FA 28 cmp edx, 40

000001FE 7C C3 jl OutputSorted

output prompt2, outputstring

00000219 B8 00000000 mov eax, 0

0000021E C3 ret

0000021F \_MainProc ENDP

END

Microsoft (R) Macro Assembler Version 12.00.31101.0 03/20/15 02:20:30

example.asm Symbols 2 - 1

Macros:

N a m e Type

atod . . . . . . . . . . . . . . Proc

atow . . . . . . . . . . . . . . Proc

dtoa . . . . . . . . . . . . . . Proc

input . . . . . . . . . . . . . Proc

output . . . . . . . . . . . . . Proc

wtoa . . . . . . . . . . . . . . Proc

Segments and Groups:

N a m e Size Length Align Combine Class

FLAT . . . . . . . . . . . . . . GROUP

STACK . . . . . . . . . . . . . 32 Bit 00001000 Para Stack 'STACK'

\_DATA . . . . . . . . . . . . . 32 Bit 0000020C Para Public 'DATA'

\_TEXT . . . . . . . . . . . . . 32 Bit 0000021F Para Public 'CODE'

Procedures, parameters, and locals:

N a m e Type Value Attr

\_MainProc . . . . . . . . . . . P Near 00000000 \_TEXT Length= 0000021F Public

outerLoop . . . . . . . . . . L Near 00000024 \_TEXT

innerLoop . . . . . . . . . . L Near 00000044 \_TEXT

done1 . . . . . . . . . . . . L Near 00000056 \_TEXT

Copy . . . . . . . . . . . . . L Near 000000D2 \_TEXT

OutLoop . . . . . . . . . . . L Near 000000FF \_TEXT

InLoop . . . . . . . . . . . . L Near 00000105 \_TEXT

Continue . . . . . . . . . . . L Near 0000010D \_TEXT

MinValue . . . . . . . . . . . L Near 00000138 \_TEXT

PrintResults . . . . . . . . . L Near 00000142 \_TEXT

OutputOriginal . . . . . . . . L Near 0000014C \_TEXT

OutputSorted . . . . . . . . . L Near 000001C3 \_TEXT

Symbols:

N a m e Type Value Attr

@CodeSize . . . . . . . . . . . Number 00000000h

@DataSize . . . . . . . . . . . Number 00000000h

@Interface . . . . . . . . . . . Number 00000000h

@Model . . . . . . . . . . . . . Number 00000007h

@code . . . . . . . . . . . . . Text \_TEXT

@data . . . . . . . . . . . . . Text FLAT

@fardata? . . . . . . . . . . . Text FLAT

@fardata . . . . . . . . . . . . Text FLAT

@stack . . . . . . . . . . . . . Text FLAT

\_getInput . . . . . . . . . . . L Near 00000000 FLAT External

\_showOutput . . . . . . . . . . L Near 00000000 FLAT External

atodproc . . . . . . . . . . . . L Near 00000000 FLAT External

atowproc . . . . . . . . . . . . L Near 00000000 FLAT External

count1 . . . . . . . . . . . . . DWord 000001DC \_DATA

count2 . . . . . . . . . . . . . DWord 000001E0 \_DATA

dtoaproc . . . . . . . . . . . . L Near 00000000 FLAT External

label1 . . . . . . . . . . . . . Byte 000001D3 \_DATA

outputstring . . . . . . . . . . Byte 000000CE \_DATA

prompt1 . . . . . . . . . . . . Byte 00000000 \_DATA

prompt2 . . . . . . . . . . . . Byte 00000025 \_DATA

scoreArray . . . . . . . . . . . DWord 00000082 \_DATA

scoreString . . . . . . . . . . Byte 00000031 \_DATA

sorted . . . . . . . . . . . . . DWord 000001E4 \_DATA

temp2 . . . . . . . . . . . . . Byte 000000C2 \_DATA

tempx . . . . . . . . . . . . . Byte 000000B6 \_DATA

temp . . . . . . . . . . . . . . Byte 000000AA \_DATA

wtoaproc . . . . . . . . . . . . L Near 00000000 FLAT External

0 Warnings

0 Errors

