

Note: Keep in mind that this review is designed as a practice exam & is not intended to be comprehensive. While studying for the exam, it would be prudent to consider variations on the questions presented here. You should also include the homework questions when reviewing for the exam. Finally, it is possible to get strong clues for answering some of the questions in the review by referring to other questions. In general, you should not depend on in the actual exam.

1. What information does a stack frame contain? Why do we use it to pass arguments to procedures?

2. Compare & contrast C calling convention with STDCALL convention.

3. Compare & contrast a global variable with a local variable created on the stack.

4. Write some assembly code for a procedure which determines the minimum element of an array of DWORDs, passing all information into & out of the procedure using a stack frame. No registers should be visibly modified by the procedure; that is the procedure should save & restore any register it might use.

Stack before procedure is called:

<u>address</u>	<u>contents</u>
???	???
???-4	???
???-8	ArrayOffset
???-12	ArraySize ◀ (ESP)

Stack after procedure returns:

<u>address</u>	<u>contents</u>
???	???
???-4	???
???-8	ArrayOffset
???-12	ArraySize
???-16	min element ◀ (ESP)

5. Consider the following code listing; the right most column indicates the line's location in memory. Assume the stack starts out empty & diagram the stack (including ESP) where indicated. Note you may have more rows than you really need.

```

main PROC
    00000000    call Clrscr

    00000005    pushd 0abcdh
    0000000A    pushd 0efefh
    0000000F    call  procX
    00000014    pop     ECX
; C) diagram stack here
    exit
main ENDP

procX PROC
; A) diagram stack here
    0000001C    push  EAX
    0000001D    push  EBX

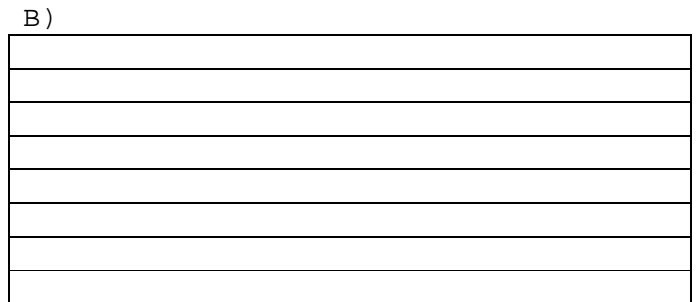
    0000001E    mov   EAX, [ESP+12]
    00000022    mov   EBX, [ESP+16]
    00000026    cmp   EAX,EBX

    00000028    jl   skip
    0000002A    mov   [ESP+16],EAX

skip:
    0000002E    mov   EAX,[ESP+8]
    00000032    mov   [ESP+12],EAX
    00000036    pop   EBX
    00000037    pop   EAX
    00000038    add  ESP,4
; B) diagram stack here
    0000003B    ret
procX ENDP

END main

```



6. Describe the function of the LEA instruction. Why do we need it in addition to the OFFSET directive?

7. Diagram a typical input port. What are the possible outcomes if we attempt to write to it?

8. Build a Huffman Encoding tree based on the following instructions & frequencies:

mov	60%
add	10%
inc	8%
jmp	7%
jle	5%
push	4%
mul	3%
pop	2%
rot	1%

9. What does DMA stand for? What is the primary motivation for selecting DMA as an I/O mechanism?

10. Describe a strength & a weakness of the File Allocation Table scheme.

11. Describe a strength & a weakness of the List of Blocks File Organization scheme.

12. What factors (tradeoffs) must be considered when designing an instruction set?