Hints and tips:

* Most of the questions in the final are closer to midterm 2 than midterm 1. This is because I don’t need to ask you specifically how data look like in memory or how to convert from binary to hex, these topics are built-in to the midterm 2 type of questions.
* The T/F questions are on Modules 1, 2, 3. They are to catch the “theories” that can’t be easily shown by coding (within 2 hours), such as cache, instruction cycles, memory management, etc.
* To prepare for the final, go through the class exercises of modules 4 – 8 and the midterms without looking at the solution. If you can do them, you should be okay with the final.

The only new topics in the final are from module 8: string instructions and 2D arrays

1. T/F questions. Similar to the quizzes and midterm 1.

Example: the value 0A0h is a negative value if it’s a signed WORD.

Answer: False

Because as a word, the data is 00 A0  
and because the first bit is 0, the value is positive

2. Short answers

Example 1: Give an example of a CISC instruction from the Intel instruction set. Explain why it's a CISC instruction.

There are multiple good answers, here are 2:

scasb because it compares [edi] and AL, and increments/decrements edi

loop because it decrements ecx, compares ecx to 0, and jumps to the label if ecx is not 0

Example 2: Step through the instructions and show change in the general purpose registers.  
mov eax, -1 ; eax = FFFF FFFFh or 1111 1111 1111 1111 1111 1111 1111 1111b

shr eax, 2 ; eax = 0011 1111 1111 1111 1111 1111 1111 1111b

; or eax = 3FFF FFFFh

cdq ; edx = 0

mov ebx, 1000h ; ebx = 1000h

idiv ebx ; eax = 0003 FFFFh edx = 0000 0FFFh

; remember that divide in hex works the same way as divide in decimal:

; divide by 1000h means remove the last 3 digits

; and multiply works the opposite way: A2h \* 10h = A20h

Example 3: The user wants to do the task that's described in the comment.  
If there is error in the instructions below so that the task can't be done, correct the error.

mov eax, offset arr ; arr contains signed words

; OK:

; eax contains address of arr

mov edx,[eax] ; want: edx = first value of arr

; error

; fix:

; movsx edx, WORD PTR [eax]

mov [eax + 2], 5 ; want: arr[2] = 5

; error

; fix:

; mov WORD PTR[eax + 4], 5

3. Writing code.

Example 1: Write code to implement the following loop. All variables are unsigned bytes. Catch all possible errors with jumps to error label E1.

while (var1 > var2)

var2 = var2 \* 5

One possible answer:

mov al, var2 ; note that these 2 instructions are outside of loop

mov bl, 5 ; because they don't need to be repeated

whileLoop:

cmp var1, al ; var1 > var2 ?

jbe endLoop ; False: end loop

mul bl ; Otherwise fall through to loop body

jc E1 ; check for error

jmp whileLoop ; if no error, loop back

endLoop:

mov var2, al ; just as with the first 2 intructions

; this one is outside of the loop

E1:

Example 2: The following question is longer than the typical exam question. It could be 3 different exam questions, to give you more practice, and in the real exam you would only need to do one part of this question.

Write a procedure to copy str1 to str2. Both strings are text strings that are null terminated.

The procedure returns 1 if all of str1 can be copied to str2, and returns 0 if str1 cannot be copied to str2 because str2 is smaller in size. Use string instructions as needed.

The procedure call is

sub esp, 4 ; for return value of 1 or 0

push OFFSET str1

push OFFSET str2

call stringCopy

See the file final\_sample\_last\_question.asm