

(CSCI 365) Exam Three Review

Note: Exam Three covers Chapter 1, 2, and 3. The test will consist of two parts. Part I consists of MC/MA/TF questions, which come from the same pool of your homework. So review homework, and understand why the answer is correct, is very important. Part II consists of short answer questions. It could be coding, calculation (show work), or concepts. Make sure you understand how to do the following sample questions. The real test questions may not be the same as the sample questions. However, know how to do sample questions will improve your real test grade for sure.

Concepts from Chapter 7 as summary in author's PPT lecture

1. Shift and rotate instructions are some of the best tools of assembly language
 - finer control than in high-level languages
 - SHL, SHR, SAR, ROL, ROR, RCL, RCR
2. MUL and DIV – integer operations
 - close relatives of SHL and SHR
 - CBW, CDQ, CWD: preparation for division
3. 32-bit Mode only:
 - Extended precision arithmetic: ADC, SBB
 - ASCII decimal operations (AAA, AAS, AAM, AAD)
 - Packed decimal operations (DAA, DAS)

Concepts from Chapter 8 as summary in author's PPT lecture

4. Stack parameters
 - more convenient than register parameters
 - passed by value or reference
 - ENTER and LEAVE instructions
5. Local variables
 - created on the stack below stack pointer
 - LOCAL directive
6. Recursive procedure calls itself
7. Calling conventions (C, stdcall)
8. MASM procedure-related directives
 - INVOKE, PROC, PROTO

Concepts from Chapter 9 as summary in author's PPT lecture

9. String primitives are optimized for efficiency
10. Strings and arrays are essentially the same
11. Keep code inside loops simple
12. Use base-index operands with two-dimensional arrays
13. Avoid the bubble sort for large arrays
14. Use binary search for large sequentially ordered arrays

Sample Short Answer Test Questions

Sample Short Answer Questions (6-8 similar questions may show up in real test)

1. Write a sequence of shift instructions that cause AX to be sign-extended into EAX. In other words, the sign bit of AX is copied into the upper 16 bits of EAX. Do NOT use the CWD instruction.
2. Suppose the instruction set contained no rotate instructions. Show how you would use SHR and a conditional jump instruction to rotate the contents of the AL register 1 bit to the right.
3. Write a single rotate instruction that exchanges the high and low halves of the DL register
4. Write a sequence of instructions that shift three memory words to the left by 1 bit position. Using the following test data:
byteArray WORD 810Dh, 9c64h, 93ABh
5. Write instructions that divide -276 by 10 and store the result in a 16-bit variable val1
6. Implement the following C++ expression in assembly language, using 32-bit unsigned operands: $val1 = (val2 / val3) * (val1 + val2)$
7. Create a procedure named AddThree that receives three integer parameters and calculates and returns their sum in the EAX register
8. Declare a local variable named pArray that is a pointer to an array of doublewords
9. Declare a local variable named myArray that is an array of 20 doublewords
10. Create a procedure named SetColor that receives two stack parameters: forecolor and backcolor, and calls the SetTextColor procedure from the Irvine32 library
11. Create a procedure named WriteColorChar that receives three stack parameters: char, forecolor, and backcolor. It displays a single character, using the color attributes specified in the forecolor and backcolor.
12. Declare a procedure named MultArray that receives two pointers to arrays of doublewords and a third parameter indicating the number of array elements. Also, create a PROTO declaration for this procedure.
13. Show an example of a base-index operand in 32-bit mode
14. Show an example of a base-index-displacement operand in 32-bit mode
15. Suppose a two-dimensional array of doublewords has three logical rows and four logical columns. Write an expression using ESI and EDI that addresses the third column in the second row. (Numbering for rows and columns starts at zero)
16. Write instructions that use SCASW to scan for the 16-bit value 0100h in an array named wordArray and copy the offset of the matching member into the EAX register
17. Write a sequence of instructions that use the Str_cmp procedure to determine the larger of two input strings and write it to the console window.
18. Show how to modify the Str_ucase procedure from the Irvine32 library so it changes all characters to lower case