Practice Questions for exam 2

The following are topics likely to be covered by the second exam.

1. The stack frame. Pass parameters on the stack to a proc and access them there. Use **ret n** to clean up the stack on return. (try coding ArraySearch)
2. Proto and Invoke. Write a proc ArraySum which is passed an array’s offset and it’s size. Write the proto, invoke (and proc) lines of code.
3. Indexed array manipulation as used in our word sorting program
4. Recursion: code something recursively. ArraySearch would be at the hard end of this spectrum. Factorial at the easy end.
5. Write a macro with arguments/parameters like readString or writeString
6. Use lods/stos string functions to encrypt the ASCIIZ string stored in data at ASZSTRING. Get a random value between 0 and 255 to encrypt with. Store the encrypted string at a previously allocated byte address ENCRYPT.
7. We wrote a prime number tester but could not write it efficiently without a sqrt function. See if N (a dword in your data area) is prime using the 87 processor to calculate the sqrt of N. (Ignore rounding issues.)
8. Modify my link list which used the stack so the user could enter fixed-length strings instead of ints. User could enter ‘0’ to quit.
9. An 87 problem… Suppose you have dword int variables A,B,C,D,E. Perform the following (postfix) arithmetic using the 87 processor:

A B + C \* D E / -

(Ignore rounding issues.)

Short answer topics:

* Get time in seconds after calling the DOS interrupt code ah=2cH
* write a string in a random location
* what is mouse interrupt function code?
* Name all 8086 (regular or extended) registers.
* In a 16-bit program, how big are the code and data segments? How do you know?
* What are the string primitive instructions? Pick one and indicate the available versions/options for different datatypes.
* What would -78450098 look like stored as packed BCD? Clearly indicate exactly where the digits would go. Also label the data area appropriately for use by the 87 coprocessor.
* Cluster links as per text examples…you can see [my ppt](http://www.oneonta.edu/faculty/higgindm/assembly/Disk%20fundamentals.ppt).
* Date/Time stamp configuration… you can see [my ppt](http://www.oneonta.edu/faculty/higgindm/assembly/Disk%20fundamentals.ppt). Date stamp is a word with bits 0..4 for day, and 5-8 for month. Is this a legal date: 0011101001011111? Time stamp uses bits 0..4 for seconds, and 11-15 for hours. What time is this? 1000101100110000?
* I have shown just 8 clusters in a chain, below, numbered from 0, 1, …, 7. If a cluster is 4 k , how big is the file bob.txt?

Filename Starting cluster

Bob.txt 0

Sue.txt 1

|  |
| --- |
| 4 |

|  |
| --- |
| 3 |

|  |
| --- |
| 2 |

|  |
| --- |
| 6 |

|  |
| --- |
| 5 |

|  |
| --- |
| eoc |

|  |
| --- |
| eoc |

|  |
| --- |
| 7 |

|  |
| --- |
| 0 |

|  |
| --- |
| 1 |

|  |
| --- |
| 7 |

* IEEE single-precision f-p format (convert to or from). (This is in text chapter 17 on the -87 coprocessor.) Convert -54.26 to IEEE single precision f-p format. What value is 415A0000h in decimal, assuming it was an IEEE single precision f-p value.
* When is the **public** directive used? How about **extern**?

 What is generated by the following macros?

 1. FORC code,<@#$%^&\*!<!>>

 BYTE "&code"

 ENDM

2.

f1 = 1

f2 = 1

gen macro arg

dword arg

endm

f3 = f1 + f2

gen %f1

gen %f2

WHILE f3 LT 0FFh

 Gen %f3

 f1 = f2

 f2 = f3

 f3 = f1 + f2

ENDM

3.

iVal = 10

gen macro arg

dword arg

endm

REPEAT 10

 Gen %iVal

 iVal = iVal + 10

ENDM