

CSCI 2160
Exam 2
Study Guide Fall 2011

1. (5 pts) Suppose that you have the function whose prototype is defined as

```
int func(short sX, int iY, byte bZ)
```

Assume that all results will fit into the defined storage. (Be careful!).

```
sA    word    ?  
bB    byte    ?  
sC    word    ?  
sD    word    ?
```

Write the assembly code call sequence which would accomplish the following tasks. (I deliberately did NOT put any casting information)

```
sD = func(sA, sC, bB);
```

2. (4 pts) Suppose that you have written a program **exam2.asm** that makes reference to the function **inorder** contained in the file **btree.obj**

a. What statement(s) must be supplied in the exam2.asm file that will allow your program to assemble properly?

b. Where must the statement(s) be placed in the exam2.asm file?

3. (4 pts) Suppose that there are a set of macros in a file named **macros.asm** and you wish to use them in a program **exam2.asm** that you have written.

a. What statement(s) would you need to put in exam2.asm to do this?

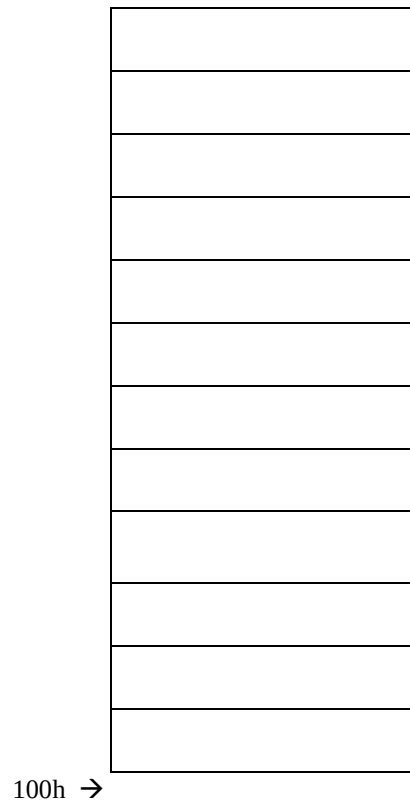
b. Where would these statement(s) be placed?

4. (6) Given the below function prototype generate the call sequence. Then, label both sides of the stack indicating how the stack pointer changes on the left and how the variables will be referenced as an offset from the base pointer on the right. Assume that you are to illustrate the stack after the `mov EBP, ESP` statement.

prototype: `int func3(short sX, int iY, int iT)`

iA dword ?
iC dword ?
sB word ?
iZ dword ?

iZ = func3 (iB, iA, iC);



5. (3)What will be the contents (in HEX) of AX and DX after the following operation? Be careful, the values are in hex.

MOV	DX,55h	AFTER:	AX: _____
MOV	AL,32h		
CBW			
CWD			BX: _____
MOV	BX,7h		
IDIV	BL		DX: _____

6. (3) What will be the contents (in HEX) of AX and DX after the following operation? Be sure to write EVERY hex digit in the register.

MOV	DX,55h	AFTER:	AX: _____
MOV	AX,32h		
MOV	CX,10h		CX: _____
IMUL	CX		DX: _____

7. (11 pts) Write the function **int findRecord(int*, int)** where **int*** is the beginning address of a 2-dimensional table where the first column is an int value representing an identifying key of a record stored in a database. The second column is an int which represents the record's number in the database (e.g. record 5, record 0, record 9). The last key in the table is -1. The function will search the table looking for the value indicated by the 2ND parameter. If the value cannot be found, the function returns -1. If the value IS found, the function returns the corresponding record number. An example of a VERY short table is given below: The first int is a key, the second int is its record number in the file, the 3rd int is a key, the 4th int is its corresponding record, the 5th is a key, the 6th is its record number, etc...

47	9	22	6	85	4	17	2	-1	-1					
key	key		key		key									

8. (3) Generate a function call using the function from problem 5 to determine the position of a record in a file whose key is **iValue**. Write your code so that the appropriate message is displayed (either found or error). You may assume that you have macro **puts** available for displaying a string.

```

iTableIndex    dword    27,2,82,4,76,5,34,6,92,7,15,83,9,105,26,-1,0
strError       db       10,13,"The value could not be found",0
strFound       db       10,13,"The value was found",0
iValue         dword    ?

```

9. (6 pts) Given the contents of the registers and data segment: EAX: FFCE3D02
 EBX: 00000E45 ESP: 0402AE28 (NOTE: YOU MUST USE ESP FOR INIT value of stack ptr)

0410 01B0 iA dword 37
 0410 01B4 iB dword 25
 0410 01B8 iC dword iB (reference to the memory location iB)

The partial contents of the stack are given below. I have only listed the last 5 hex digits of the even-numbered addresses because of space. Assume the values to the left of the variables above represent their true locations in memory.

2A	E2	40	52	6A	5D	F2	02	41	FF	2C	B2	22	53	7E	42	87	A5	29	33	AA	E4
2AE20		2AE22		2AE24		2AE26		2AE28		2AE2A		2AE2C		2AE2E		2AE30		2AE32		2AE34	

Give the results **after** execution of the below statements. Assume in **EACH** case that the registers are as originally defined above.

a) (2 pts) POP EAX EAX: _____ ESP: _____

b) (2 pts) PUSH iC ESP: _____

c) (2 pts) Show below what bytes were affected in 2) by crossing out and writing above the addresses the new values:

2A	E2	40	52	6A	5D	F2	02	41	FF	2C	B2	22	53	7E	42	87	A5	29	33	AA	E4
2AE20		2AE22		2AE24		2AE26		2AE28		2AE2A		2AE2C		2AE2E		2AE30		2AE32		2AE34	

10. (15 pts) In the below problems, write the appropriate assembly commands for each calculation. Assume the prefix indicates the size of the variable. I (int), s (short), b (byte), q (quad word).

a) qA = iD/sB

b) iF = sC * sE

c) $iB = sD \% sC \quad (\text{modulo})$

11. (6) Write a macro called **bigVal** when invoked with the statement below will put the largest of iA and iB into iC
- ```
bigVal iA, iB, iC
```

12. (4 pts) For each of the following describe every hex digit in the resulting register

```
mov bx,0C240h
```

```
movzx eax, bx EAX: _____
```

```
movsx eax, bx EDX: _____
```

13. (6 pts) Write equivalent assembly instructions for the below java code. You may NOT use the **jmp** or **cmp** instructions. You may assume that you have the below available to you in the data segment and that you have the **puts** macro available to you. ( `puts strOut` ; where strOut is a null-terminated string)

```
greeting db 10,13,"Hello World!",0

for (i = 2; i<=12; i+=2)
 System.out.print("\nHello World!");
```

14. (3 pts) Given the below data segment:

```
.data
bVal byte ?
sVal word ?
iVal dword ?
sArray word 1,2,3,4,5
iArray1 dword 10,20,30,40,50
iArray2 dword 500 dup (?)
```

Suppose that you have written:

```
lea ebx, sArray
```

Rewrite one correct assembly statement referencing ebx that will replace the value 4 with 14 in the above data area.



15. (11) Write the equivalent assembly code for the below Java code (use the back if necessary)

```
public int myFun(int N)
{
 if (N==0)
 return 1;
 if (N==1)
 return 2;
 return (myFun(N-1) + myFun(N-2));
}
```

16. (4) Write the instruction(s) to jump to bit3Off if bit 3 is off in AL (where the bits are numbered from right to left starting at 0)

17. (4 pts) Write the instruction(s) which will jump to bit2ON5OFF if bit 2 is ON and bit 5 is OFF.

18. Given the below data segment:

```
.data
bVal byte ?
wVal word ?
dVal dword ?
wArray word 1,2,3,4,5
dArray dword 10,20,30,40,50
cArray dword 500 dup (?)
```

a) (10 pts) After each of the below instructions is executed describe what is in the EAX register. Be sure to write EVERY hex digit.

|     |      |                 |            |
|-----|------|-----------------|------------|
| mov | eax, | TYPE bVal       | EAX: _____ |
| mov | eax, | TYPE wArray     | EAX: _____ |
| mov | eax, | TYPE cArray     | EAX: _____ |
| mov | eax, | LENGTHOF dVal   | EAX: _____ |
| mov | eax, | LENGTHOF dArray | EAX: _____ |
| mov | eax, | LENGTHOF cArray | EAX: _____ |
| mov | eax, | SIZEOF bVal     | EAX: _____ |
| mov | eax, | SIZEOF dArray   | EAX: _____ |
| mov | eax, | SIZEOF cArray   | EAX: _____ |
| mov | eax, | OFFSET dArray   | EAX: _____ |

19. (5) Give an `.IF` directive that can be used instead of the below assembly code.

```
 cmp eax,ebx
 jae doThis ; jae means "above or equal"
 add eax,ecx
 imul ebx
 jmp continue
doThis:
 add ebx,ecx
 mov eax,edx
 imul ebx
continue:
```

20. (3 pts each) Given the contents of the below registers (in hex). Give what will be in the register as a result of the instructions:

CX: 79 A4    sal   cx,1

CX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

AX: 89 A4    sar   ax,1

AX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

BX: E9 A4    shl   bx,1

BX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

AX: E9 A4    sar   ax,cl  
CL: 04

AX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

AX: E9 A4    rol   ax,3

AX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

AX: E9 A4    ror   ax,3

AX: \_\_\_\_\_

CF \_\_\_\_    OF \_\_\_\_

ECX: 56 78 12 34    shld  
EAX: CD EF 90 AB

ecx,eax,8    ECX: \_\_\_\_\_ CF \_\_\_\_    OF \_\_\_\_

EAX: \_\_\_\_\_

ECX: 56 78 12 34      shrd      ecx, eax, 4      ECX: \_\_\_\_\_ CF \_\_\_\_ OF \_\_\_\_  
 EAX: CD EF 90 AB  
 EAX: \_\_\_\_\_

21. (4) Given the instruction EIP register contains **001004CD** when the short jump instruction **jmp errorcond** is begun to be executed. The 2<sup>nd</sup> byte of the instruction is **C5**. What will the EIP register contain when the jmp instruction is complete?

22. (15) Given AL contains 11011011 and that BL contains 01110101 what are the results in the AL register of the following. Indicate, also, the settings of the SF and ZF :

a)      AND    AL, BL      \_\_\_\_\_      SF \_\_\_\_      ZF \_\_\_\_

b)      OR      AL, BL      \_\_\_\_\_      SF \_\_\_\_      ZF \_\_\_\_

c)      NOT    AL      \_\_\_\_\_      SF \_\_\_\_      ZF \_\_\_\_

d)      XOR    AL, BL      \_\_\_\_\_      SF \_\_\_\_      ZF \_\_\_\_

e)      NEG    AL      \_\_\_\_\_      SF \_\_\_\_      ZF \_\_\_\_