Score:

CSI 2334 Introduction to Computer Systems
Exam #2

Name: ______________________________________ Section: ___________

True / False (3 point each)

___(T) 1. Assuming AX = FFC2h, the following sequence:

```
cmp     ax,-62
jae     TEST
```

will branch to TEST.

___(F) 2. After executing the sequence of instructions given in #1, the EFLAGS bits will be:

SF	CF	OF	ZF
0	1	0	0

___ (F) 3. Assuming ebx is pointing to an array of 16-bit values, and ecx holds the number of values to process, the following piece of code will convert negative values stored in the array to positive:

```
        mov     bx,-1
TOP:    mov     ax,[ebx]
        cmp     ax,0
        jge     NEXT
        cwd
        idiv    bx
        mov     [ebx],ax
NEXT:   add     ebx,2
        loop    TOP
```

____(F) 4. When we use **pop** to retrieve a value from the stack, it will post-increment the stack pointer, **then** move what the stack pointer is pointing at into the destination.

____(F) 5. The machine language encoding for **mov [eax], bx** is **89 18h**

Multiple Choice (3 point each)

____6. Which of the following statements about the `__declspec()` C++ calling convention are true?

a. In a C++ function call, the parameters are widened to 32 bits and passed in reverse order.
b. C++ pushes the current contents of EIP after it has pushed the parameters.
c. After function termination, C++ cleans up the stack, restoring the stack pointer to its location before the function prologue.
d. all of the above
e. none of the above

____7. Which assembly language instruction pushes the address of the next instruction onto the stack and then transfers control to procedure code?

a. call
b. ret
c. pushad
d. pushfd
e. ret 4

____8. Which of the following is **not** a directive:

a) `input prompt,string,40`
b) `.STACK 4096`
c) `mov eax`
d) `.DATA`
e) a and c above

____9. The following instruction **will** need a prefix byte while working on a 32-bit system:

```
add ax, wordOp
```

a) Yes
b) No

____10. How do I correctly establish a pointer to an array of characters called `buffer` if C++ is passing `buffer` as the only parameter to my assembly language function, and I have NOT established a stack frame?

a. `mov ebx,[esp+4]`
b. `mov ebx,[esp+8]`
c. `lea ebx,buffer`
d. `mov ebx,[ebp+8]`
e. none of the above

Short Answer

11. (15 points) Assume that the following commands execute sequentially. Indicate the results of each instruction in hex and the resulting changes in the EFLAGS register after each. Assume that the first set of values under the EFLAGS bits are the values before the first instruction.

			SF	ZF	CF	OF
			1	0	0	1
`mov`	`eax,71`	`EAX: 0000 0047`	1	0	0	1
`mov`	`ebx,-4`	`EBX: FFFF FFFC`	1	0	0	1

`add`	`eax,ebx`	`EAX:0000 0043`
		`EBX:FFFF FFFC`
			0	0	1	0

`cdq`		`EDX:0000 0000`
		`EAX:0000 0043`
			0	0	1	0

`idiv`	`ebx`	`EDX: 0000 00003`
		`EAX: FFFF FFF0`
		`EBX: FFFF FFFC`
			?	?	?	?

`cmp`	`edx,0`	`EDX: 0000 00003`	0	0	0	0
`je`	`CONTINUE`	`Will you jump to CONTINUE?`	yes/no			

Work Area:

12. (15 points) Write an assembly language function definition that will search an array of ten 16-bit values to determine whether the key is in the list or not and return the index where it was found (if found). You may assume you have one global variable which defines the maximum size of array. Establish stack frame for your solution. Include comments in each line of code. The AL function call and function prototype are below:

function call: `asmSearch(myArray, key, index)`

function heading: `void __declspec(naked) asmSearch(const short[], short, short&)`

```
const short MAX = 10;                     // max size of array


void __declspec(naked) asmSearch(const short[], short, short&)
{
  __asm
  {
  START:

      push ebp                          //preserve previous base pointer
      mov ebp, esp                      // establish stack frame

      movsx    ecx, MAX                 // init CD ctr to MAX
      jecxz    DONE                     // if CDctr=0 we're DONE

      mov      ebx, [ebp + 8]           // establish ptr to myArray
      mov      dx, [ebp + 12]           // move key to scratch

TOP :
      cmp      dx, [ebx]                // compare key to current value
      je       DONE                     // if key = curr val we're DONE
      add      ebx, 2                   // otherwise, move to next array element
      loop     TOP                      // decr CDctr, if !=0 do TOP again!

DONE :
      sub      cx, MAX                  // calculate index
      neg      cx
      mov ebx, [ebp + 16]
      mov [ebx], cx                      // store index in memory

      mov esp, ebp                      //deallocate local variables
      pop ebp                           //restore previous ebp

      ret                               // return to calling code
  }
}
```

13. (5 points) Suppose `eax` holds a value and some action needs to be taken when that value is larger than 100. Which one is the appropriate instruction if the value is signed and what problems will occur if used otherwise? Explain briefly.

```
cmp eax, 100
 ja bigger
```

or,

```
cmp eax, 100
 jg bigger
```

**more on page 137

14. (10 points) For each part of this problem, assume "before" values when te given instruction is executed. Give the requested "after" values. Denote carry flag and overflow flag (if applies).

	Before	Instruction	After	CF OF
a.	AX: FF75 byte at count: FC	`idiv count`	AX: FD 22	
b.	AL: 0F BH: 4C	`imul bh`	AX: 04 74	1 1

c.	ESP: 00 63 FB 60
	EBX: 22 33 44 55
	ECX: 66 77 88 99
		`push ebx`
		`push ecx`
			ESP: 00 63 FB 58
			
			EBX: 22 33 44 55
			
			ECX: 66 77 88 99
				Draw stack in work area

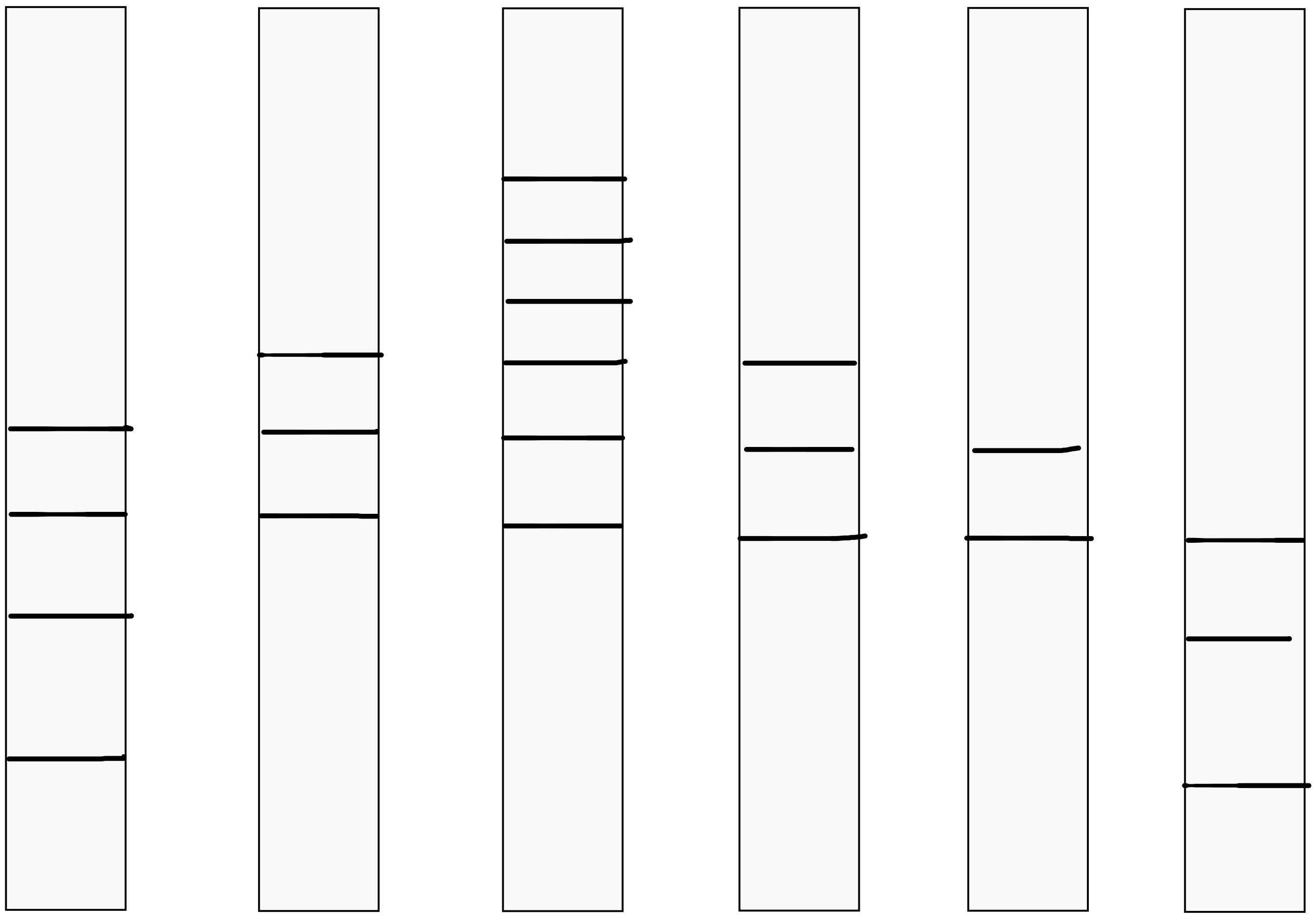
Work Area:

15. (15 points) For the function call and function heading given, draw stack pictures at the following points:

a. after the function call is executed,
b. after the stack frame is established
c. after the registers that will be used in the function (ebx, ecx, and eflags) have been preserved
d. after the registers in use (ebx, ecx, and eflags) in the function have been restored
e. after the base pointer is restored
f. after the return to the calling code has been issued

As always, you may assume the function prologue has taken place, and that the compiler will execute the function epilogue. Show all pointers to the stack, as well as the contents to which they point.

```
Function Call:    asmFunc(p1,p2,p3);
Function Heading: void __declspec() asmFunc(short,short[],short&)
```



**The structure is here, diy the rest of it

16. (10 points) *Happy Thanksgiving!*

Extra Credit

(4 points) Why is it important to minimize the number of jump instructions required to develop a solution? What is a design technique to help minimize jumps?

Violates pipelining
High level implementation design before coding

(6 points) Suppose number contains the number of times a loop body is to be executed. What will happen this backward for loop is executed? If any problem occurs, how can you fix the problem?

```
           mov ecx, number    ; number of iterations
forIndex:  .                  ; loop body
           .

           loop forIndex      ; repeat body times
```

**page 151