

1. (25 pts) _____

Name: _____

2. (30 pts) _____

This examination has five problems, each with approximate points indicated. The maximum possible total is 150 points.

3. (40 pts) _____

Show all your work. You may use the ECE 390 Laboratory Notes, one page of your own notes, and an unnecessary calculator.

4. (30 pts) _____

5. (25 pts) _____

Problem 1. (25 pts) Write short sequences of instructions to accomplish each task. You may use registers `AX`, `BX`, `CX`, `DX` without saving their contents. No comments are necessary.

(a) Memory bytes labeled `P` and `Q` contain unsigned integers p and q respectively. If $p < q$, then set the memory byte labeled `F` to 1. Otherwise, do not change `F`. Assume `F` is already declared with `RESB`.

(b) Memory words labeled `U` and `V` hold values u and v respectively in two's complement representation. If u is divisible by v (with zero remainder), then jump to label `.yes`.

Problem 3. (40 pts) You have used Turbo Debugger to suspend the execution of a program. The last two instructions executed were `call Subr` and `pusha`.

```
#exam1#Subr
cs:0357 60          pusha          sp 01EE
cs:0358▶BB0000     mov          bx,0          ds 13BC
#exam1#.loop
cs:035B 81FB0200   cmp          bx,2          es 138C
cs:035F 770F       ja          #exam1#.done   ss 139C
cs:0361 89DE       mov          si,bx         cs 13BC
cs:0363 D1E6       sal          si,1          ip 0358
cs:0365 8A840001   mov          al,[si+0100]  ss:0204 2074
cs:0369 88870C01   mov          [bx+010C],al  ss:0202 6168
cs:036D 43         inc          bx            ss:0200 5700
cs:036E EBEB       jmp          #exam1#.loop  ss:01FE 0291
#exam1#.done
cs:0370 BA0C01     mov          dx,010C       ss:01FC 1969
cs:0373 E83650     call         dspmsg        ss:01FA 1308
cs:0376 61         popa          ss:01F8 1867
cs:0377 C3         ret           ss:01F6 1406
                                ss:01F4 01FE
                                ss:01F2 2004
ds:0100 42 4C 55 45 47 52 41 53  BLUEGRAS  ss:01F0 1984
ds:0108 53 20 4D 55 53 49 43 24  S MUSIC$  ss:01EE 01F2
```

(a) Express each answer in hexadecimal. If there is insufficient information, write “Unknown.”

(i) The 20-bit linear address of the next instruction to be executed: _____

(ii) The value of the word at `ds:010F` at this time: _____

(iii) The value of the word at `ss:01F7` at this time: _____

(iv) The value of the `CX` register at this time: _____

(b) Now suppose the execution of `Subr` continues from this point.

(i) How many times is the `cmp bx,2` instruction executed? _____

(ii) Determine the values of the flag bits immediately after the last execution of `cmp bx,2`:

SF = _____ CF = _____ ZF = _____ OF = _____

(iii) What is printed out with the call to `dspmsg`? _____

(iv) What is the value of the `IP` register immediately after `ret` is executed? _____ (hex)

Problem 4. (30 pts) This subroutine should calculate the multiplicative inverse of a number b modulo m by trying every c up to $m-1$. All numbers are unsigned integers in $\{0, 1, \dots, 255\}$.

Inputs (BL) = b
 [mvar] = byte m in memory
 Output (CL) = c such that $bc \bmod m = 1$, or 0 if b does not have a multiplicative inverse

Although there are no syntax errors, subroutine `MultiInv` has six mistakes. Identify these mistakes; for each, explain the mistake and specify a correction, using the line numbers.

```

1  MultiInv
2      CMP  BYTE [mvar], 0 ; If m=0, exit immediately
3      JE   .exit          ; to avoid division by zero
4      PUSHA
5  .loop:
6      MOV  AL, CL         ; Candidate c is in CL
7      MUL  BL             ; Compute bc
8      DIV  BYTE [mvar]   ; mod m
9      CMP  AL, 1         ; If remainder = 1
10     JE   .exit         ; then inverse has been found
11     INC  CL             ; Try next candidate c
12     CMP  CL, mvar      ; provided that
13     JL   .loop        ; c is less than m
14     MOV  CL, 0         ; Here if no inverse found
15  .exit:
16     POPA
17     RET

```

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

Problem 5. (25 pts) Write your answers in clear English.

(a) The jump condition for JGE is $SF \text{ xor } OF=0$. Show that $SF=0$ would be the incorrect jump condition by using the example `CMP AH, BH` with $(AH)=7Fh$ and $(BH)=FFh$.

(b) Explain all differences between the following NASM statements:

(1) `ADD WORD [zvar], diff` (2) `zvar RESW zvar+diff`

(c) Show that for every positive integer n , there is no multiplicative inverse of 2 modulo $2n$. Use the definitions of “multiplicative inverse” and “mod,” and give an algebraic argument. (Hint: Suppose, to the contrary, k is the multiplicative inverse of 2 modulo $2n$.)
