Name: _____

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing. In order to get full credit, **SHOW YOUR WORK**.

All code is assumed to be run on our Linux PCs in CSIF. Suppose the linker assigns the .data and .text segment addresses as in the example in the supplement to our PLN unit on machine language.

1. (5) Give a single Intel assembly language instruction which adds 1 to the word pointed to by EAX.

2. (10) Give a single Intel assembly language instruction which replaces c(EAX) by c(EAX) mod 4.

3. Look at the program at the top of p.27 (and the bottom of p.26) in our PLN unit on OS. When the **.s** file is run through **as -a**, the assembler shows the contents of the first 24 bytes of the **.text** segment to be

```
B804000000BB01000000B900000000BA04000000CD80B801
```

(a) (10) Deduce the op code for the INT instruction.

(b) (5) In the list of 24 bytes above, the linker will change bytes number _____ through _____ will change to _____. (Count the B8 as byte 0, the 04 as byte 1, etc. Fill the last blank in order of increasing address.)

4. (15) Consider the program on p.13 of our PLN unit on subroutines. Suppose the program is assembled and linked, and then run under GDB. The GDB command **x/6b $esp** will print out the hex values of the first six bytes starting at ESP, in order of increasing address. Show the numbers which will be printed out, just before the call.

5. Consider the source files **TryAddOne.c** and **AddOneMIPS.s** on p.5 of our PLN unit on MIPS/RISC. A number of modifications to the file **AddOneMIPS.s** are listed below. In each case, suppose only *that* modification is made, and state what the result of the change would be. Your choices are (i) assembly-time error, (ii) link-time error, (iii) run-time seg fault, (iv) no error announced but output would be erroneous, or (v) no errors at all.

(a) (5) all instances of $4 are replaced by $5

(b) (5) all instances of $2 are replaced by $5

(c) (5) the $31 is replaced by $13

6. (15) Consider the Java method

```
public int gy(int x)
{  int y[5],z;
   y = new int[5];
   z = x + 2;
   y[3] = x;
   ...
}
```

Show JVM assembly language which the compiler could produce from the last two assignment statements. Show both instructions and their offsets. Hint: The first two instructions for the entire function are

```
0 iconst_5
1 newarray int
```

followed by one of the instructions in the **aload** family at offset 3.

7. (10) Two arrays which are written to by the OS and read from by the hardware are the _____ and the _____.

8. (15) Assume the setting of Section 5.5 of our PLN unit on OS. Suppose the CPU has a direct-mapped cache, with line size 512 bytes. Consider the example instruction **movl $6, 8195**. In checking the cache for the destination operand, the CPU will look in either line number _____ or line number _____.

Solutions:

1.

```
incl (%eax)
```

2.

```
andl $00000003, %eax
```

3.a CD

3.b 11, 14, 94 90 04 08

4. The address of **fmt** is 0x08049094+0x14, and c(EBX) = 26 = 0x1a. So, the bytes printed out are a8 90 04 08 1a 00.

5.a Register $4 contained the address of **x**, and was used in the subroutine to reference **x**. If we use $5 instead, we'll access an unpredictable memory location, most likely a location we don't have read/write access for (probably location 0). So, the most likely outcome is a seg fault.

5.b $2 is used for "scratch paper," and $5 will perform the same role, so no problem.

6. Note that this is an instance function, so slot 0 contains **this**.

```
 0 iconst_5
 1 newarray int
 3 astore_2
 4 iload_1
 5 iconst_2
 6 iadd
 7 istore_3
 8 aload_2
 9 iconst_3
10 iload_1
11 iastore
```

7. page table, interrupt vector table

8. If the cache is addressed virtually, we use 8195 as the index.

$$\lfloor \frac{8195}{512} \rfloor = \lfloor \frac{2 \cdot 2^{12} + 3}{2^9} \rfloor = 16$$

Assuming a 32-line cache (or show it for n-line), the line number would be 16 mod 32 = 16.

In the physically addressed case, use 20483 as the index, with a result of line number 40 mod 32 = 8.