**Problem 1: Short Answers (25 points)**

**a) (5 points)** What is the output of the following code sequence?

```
int *ptr = 0x1050;
printf ("%x\n", ptr--);
printf ("%x\n", ptr);
```

**b) (5 points)** What are the three basic instruction formats for MIPS assembly? Give an example of a true assembly language (TAL) instruction for each of the formats.

**c) Pseudoinstructions (15 points, 5 points each)**
Pseudoinstructions are not actually part of the MIPS instruction set, but are often used to when writing MIPS assembly programs. These pseudoinstructions are then translated by the assembler into a sequence of "true" MIPS assembly instructions. For each pseudoinstruction below, produce the shortest possible sequence of true MIPS instructions. You can only use one additional register $at to store temporary values.

1) pseudoinstruction: swap $s0, $s1
description: $s0 = $s1, $s1 = $s0

2) pseudoinstruction: bgt $s0, num, Label
description: if($s0 > num) goto Label, where "num" is a 32 bit constant
Note: for convenience, the bit fields of num can be referenced as num[31:16] = most significant 16 bits, num[15:0] is least significant 16 bits

3) pseudoinstruction: quad $s0
description: $s0 = $s0 x 4

**Problem 2: Number Representation (20 points)**
You are designing a counter to be integrated into a digital system. The counter indicates the number of events created on the input line. You only have room to store 3 bits, therefore the counter can record a maximum of 8 events.

You are given the task of designing the number representation for the counter. You are free to choose any number representation for the counter. However, you need to minimize the power dissipation of the counter because it is used in a battery operated digital system. The switching activity of the counter is the dominate factor in power consumption.

For example, a 2-bit counter using normal number encoding, where the bit representation '00' corresponds event number 0, '01' corresponds to event number 1, and so on…

| Event Number | 2-Bit Representation |
|---|---|
| 0 | 00 |
| 1 | 01 |
| 2 | 10 |
| 3 | 11 |

This representation will have 4 bit switches when incrementing from 0 to 3:
00 -> 01: 1 bit change
01 -> 10: 2 bit changes
10 -> 11: 1 bit change

The reset takes an additional 2 bit switches (11 -> 00).

The optimal representation will have 4 bit switches when incrementing from 0 to 3 and then resetting.

Choose a number representation for the 3-bit counter that minimizes the switching activity (and therefore, the power consumption) of the counter

| Event Number | Bit Representation |
|---|---|
| 0 | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

How many bit switches does your representation require to increment from 0 to 7 and then reset?

**Problem 3: (25 points) C Functions**

You must write the code for the C function:

*char * insertChar(char *s, char c, int pos)*,

which returns the result of inserting the given character at the given position. *s* is a pointer to the first element of a string and *c* is the character that you should insert at the *pos* element of *s*. You must return the pointer to the first element of a **new** string, which is identical to *s* except with one additional character inserted at the specified position. You are not allowed to modify the string *s*. You are free to use any of the functions in the string.h library such as strlen, strcat, strcpy, etc.

**Problem 4 (30 points) Assembly**
You are to write MIPS assembly code for the following function, which uses this node structure:

```
struct node {
char name[12];
int value;
};
```

```
void exam (struct node **to) {
exam2 (*to);
(*(to-1))--;
}
```

**a) (5 points)** How many bytes of memory are required for one node structure?

**b) (25 points)** Write the MIPS assembly code for the *exam* function. You are allowed to use MAL instructions.