CS61C Summer 2001
Midterm #1
Professor Woojin Yu

Problem #1

Rewrite the following C source code using only the MIPS Assembly instructions. Please follow register conventions mentioned in class.

```
int func1(int *a)
{
        int *temp;
        if (*a)
        {
                temp = a;
                a++;
                return(*temp+func1(a));
        }
        else
                return 0;
}
```

Problem #2

In the following, some of the statements are incorrect or illegal; cross out any such bad statements. Show in the spaces provided what the remaining print statements will print when the program is executed. Briefly state why the illegal statements are wrong.

```
int main()
{
        char a = 'A', ur[] = "ORDINALS", b = 'C';
        char *alpha = &a, *beta = alpha, *gamma = ur;
        char **aleph = &alpha, **beth = β


        printf("%c\n", *gamma);


        printf("%s\n", b);


        printf("%c\n", *beta);


        printf("%c\n", alpha);
```

```
        alpha = ur + 1;

        printf("%c\n", *(ur + 3));

        printf("%s\n", &ur[1]);

        ur = alpha;

        printf("%c\n", *beta);

        beth = &(alpha);

        if ( (*aleph)[1] == (*beth)[1] )
                printf("CH is true.\n");

        Else
                printf("CH is false.\n");

        return0;
}
```

Problem #3

Convert this MIPS machine code into MAL (MIPS Assembly Language) instructions. Your final answers should use the register names, not the numbers(i.e. $t0, not $8) Also, values which represent addresses (if any) should be converted into the full 32 bit address.

ADDRESS:	Instructions:					
0x10001A00	001000	11101	11101	11111	11111	111100
0x10001A04	101011	11101	11111	00000	00000	000000
0x10001A08	000011	00000	10000	01000	00000	000111
0x10001A0C	000000	00010	00010	00010	00000	100000
0x10001A10	100011	11101	11111	00000	00000	000000
0x10001A14	001000	11101	11101	00000	00000	000100
0x10001A18	000000	11111	00000	00000	00000	001000

Problem #4

Short Answer Questions
a. Assume an 8-bit tw's complement machine on which all operations are performed on 8-bit registers. Answer the results of the following operations in hexadecimal. Assume that subtraction is done with SUBU and addition is done with ADDU.

```
        (i)     43 (hex)
              - 4A (hex)
                --------

        (ii)    82 (hex)
              + AB (hex)
                --------
```

b. List the two values that can change on exectuion of the JAL instruction.

c. Describe how the calculation of the target address for the BEQ instruction is different from that of the J instruction.

d. What output would typically be seen from running the following(correct) program on a 32-bit machine, such as the MIPS machine we are studying? The SIZEOF operator determines (at compile-time) the size (in bytes) of the type yielded by its argument.

```
#include

int main (void)
{
        char a[] = "foobar", b[15] = "baz", *c = "garply";
        int d[5] = { 1, 2, 4, 8, 16 };
        void bar (char*, int[]);

        printf("%d\n", sizeof(a));
        printf("%d\n", sizeof(b));

        bar(c, d);

        return 0;
}

void bar (char *c, int d[])
{
        printf("%d\n", sizeof(c));
        printf("%d\n", sizeof(d));
}
```

List, in order from first to last, the four values that appear.
