CS 60B Final — May 18, 1993

Your name ______________________________

login c60b–________

Discussion section number ________

TA's name ______________________________

This exam is worth 30 points, or 30% of your total course grade. The exam contains seven questions.

This booklet contains nine numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper. This is an open book exam.

When writing procedures, don't put in error checks. Assume that you will be given arguments of the correct type.

Our expectation is that many of you will not complete one or two of these questions. If you find one question especially difficult, leave it for later; start with the ones you find easier.

1	/3
2	/3
3	/4
4	/4
5	/5
6	/5
7	/6
total	/30

Question 1 (3 points):

(a) Convert the decimal number 1082 into hexadecimal.

(b) Some early computers, in order to cope with very unreliable primitive hardware, represented decimal digits in a system called *biquinary* in which each digit was encoded using five bits, exactly two of which were 1, and the other three 0. If the five bits are numbered 0, 1, 2, 4, and 7, then every decimal digit can be represented using two bits whose values add up to the desired digit (except 0, which is represented as 4+7):

	7	4	2	1	0		7	4	2	1	0
0	1	1	0	0	0	5	0	1	0	1	0
1	0	0	0	1	1	6	0	1	1	0	0
2	0	0	1	0	1	7	1	0	0	0	1
3	0	0	1	1	0	8	1	0	0	1	0
4	0	1	0	0	1	9	1	0	1	0	0

(These ten codes are the only possible biquinary codes.) The idea was that any hardware malfunction would probably produce a code that didn't have exactly two 1 bits, which could then be detected by checks in the hardware.

Lem E. Tweakit wants to know whether he could use *bioctal* (two 1 bits and six 0 bits) to represent a capital or lower case letter in a byte. Help him by telling us exactly how many different bioctal codes are possible.

(c) What is the largest value that can be used as the immediate operand in a MIPS hardware `ADDI` instruction?

Your name ____ login c60b–_______

Question 2 (3 points):

True or false:

(a) The MIPS processor automatically disables interrupts when an interrupt occurs, without any software assistance.

(b) The MIPS processor automatically enables interrupts when the interrupt routine jumps back to the user program, without any software assistance.

(c) An output device can interrupt even when it has no data to output.

(d) In the `mips4` assignment, we needed to disable the interrupt bit in the Receiver Control Register when the input buffer was full so that new characters being input would not be lost.

(e) The interrupt routine must check the ready bit in each device Status Register and loop until the device turns the ready bit on.

(f) All pages in the MIPS virtual memory system are the same size.

Question 3 (4 points):

Translate the following fragment of C code into MIPS assembler:

```
int SpaceCount(char *str) {
    int result=0;
    char ch;

    while ((ch = *str++) != '\0')
        if (ch == ' ')
            result++;
    return result;
}
```

Your name ____ login c60b–________

Question 4 (4 points):

Given the following definitions:

```
struct node {
    int value;
    struct node *next;
}

typedef struct node *List;

#define NIL ((List)0)
```

translate the following fragment of C code into MIPS assembler:

```
void delete(int num, List p) {
    while (p->next != NIL) {
        if (p->next->value == num) {
            p->next = p->next->next;
            return;
        } else
            p = p->next;
    }
}
```

(This code assumes that the list begins with a dummy node.)

Question 5 (5 points):

Write a C function `PigLatin` that takes a character string as its argument and returns its translation into Pig Latin. You may assume that the argument string contains only lower case letters (a to z). To translate a word into Pig Latin, move all the consonants that come before the first vowel to the end of the word, then add “ay”; the word “spring” should be translated to “ingspray” because the initial consonants “spr” are moved to the end. If there are no vowels (a, e, i, o, u) in the argument word, your function should return NULL.

Return a newly allocated string, not a static array.

Your name ____ login c60b–_______

Question 6 (5 points):

You are working with binary trees, like the ones in the last project, constructed out of nodes like this:

```
struct node {
    char *answer;       /* nonzero for leaf node */
    char *question;     /* nonzero for branch node */
    struct node *yes;
    struct node *no;
};
```

Write the C procedure `depth` that takes a pointer to a tree as its argument. It should return the depth of the tree—the longest path from the root node to a leaf node. (If the tree has only one node, so the root *is* a leaf node, the answer is 1. If the root node is a branch, and *both* children are leaves, the answer is 2.)

Question 7 (6 points): Find, *explain*, and *fix* the bugs in the following programs. Each program has exactly one bug. (You may need to change more than one line to fix the bug, however.)

(a) A procedure to store the null string in a character string variable:

```
void EraseString(char *s) {
    s = "";
}

/* sample invocation: */

    char text[30] = "initial value";
    ...
    EraseString(text);
```

(b) This procedure adds a new node at the end of a list of nodes. **Assume that the list already contains some nodes.**

```
struct node {
    int value;
    struct node *next;
};

void append(struct node *newnode, struct node *list) {
    struct node *p = list;

    while (p != NULL)
        p = p->next;
    p->next = newnode;
    newnode->next = NULL;
}
```

Question 7 continues on the next page.

Your name ______________________________________ login c60b–_______

Question 7 continued.

(c) This procedure creates strings with many copies of the same character, like “rrrrrrr”:

```
char *copies(int howmany, char ch) {
    int i;
    char *s;

    for (i = 0; i < howmany; i++)
        s[i] = ch;
    return s;
}
```