CS 60B Midterm #2 — November 1, 1993

Your name ____

login c60b–_______

Discussion section number _______

TA's name ____

This exam is worth 15 points, or 15% of your total course grade. The exam contains four substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly and put your name and login correctly at the top of each of the following pages.

This booklet contains five 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, write straightforward code. Do not try to make your program slightly more efficient at the cost of making it impossible to read and understand.

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.

0	/1
1	/3
2	/3
3	/4
4	/4
total	/15

Question 1 (3 points):

Without using any global variables, write a C function `previous` that takes one integer argument. The first time it's invoked, it should return zero. The next time, it should return whatever argument was given the first time. Each time thereafter, it should return the argument from the previous invocation. For example:

```
previous(23)   -->  0
previous(4)    -->  23
previous(-6)   -->  4
previous(87)   -->  -6
```

Your name ____ login c60b–_______

Question 2 (3 points):

(a) Given the following type declarations:

```
struct a {
    int i;
    double d;
};
union b {
    int i;
    double d;
};
```

What is `sizeof(struct a)`? What is `sizeof(union b)`?

(b) The following C procedure is supposed to take a pointer to a character string and modify the caller's pointer so that it points to the first nonspace. The procedure has a bug. Explain and fix it.

```
void skipspace(char *p) {
    while (*p == ' ') p++;
}
```

(c) The following program is supposed to read and print lines using a globally allocated buffer. The program has a bug. Find and fix it.

```
#include <stdio.h>
extern char buffer[100];

main() {
    while (fgets(buffer, 100, stdin) != NULL)
        fputs(s, stdout);
}
```

Question 3 (4 points):

Consider the following C procedure:

```
int longword(char *str) {
    char ch;
    int length = 0, other;

    while (*str == ' ') str++;
    while ((ch = *str++) != ' ' && ch != '\0') length++;
    if (ch == '\0') return length;
    other = longword(str);
    if (other > length) return other;
    return length;
}
```

This procedure takes a character string as its argument, and returns the length of the longest word in the string.

Your job is to translate this procedure into MIPS assembler. You may use assembler pseudo-instructions if you want. Use registers `$16` for `ch`, `$17` for `length`, and `$18` for `other`. (Of course `str` goes in `$4` and the value is returned in $2.)

The procedure shown above is not the only possible algorithm for this problem. Do not invent a different one! Translate the C procedure shown above into MIPS assembler.

Your name ____ login c60b–________

Question 4 (5 points):

We are implementing linked lists of integers using the following definitions:

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

typedef struct node *List;

#define NIL (List)0
```

Write a C function `interleave` that takes two `List`s as its arguments. It should return a single list formed by rearranging the pointers so that the elements from the two lists alternate in the result. For example, if the first list is (in Lisp notation)

```
(3 47 2 18 21)
```

and the second list is

```
(9 -5 100 14 87 92 5)
```

then the result should be

```
(3 9 47 -5 2 100 18 14 21 87 92 5)
```

Notice that if one list ends before the other, you shouldn't lose the extra elements of the longer one. Do not allocate new nodes; modify the argument lists.

Write your answer on the back of this page!