

Your name _____

login c60b-_____

Discussion section number _____

TA's name _____

This exam is worth 12 points, or 12% of your total course grade. The exam contains five 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 four 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	/1
2	/1
3	/1
4	/4
5	/4
total	/12

Question 1 (1 point):

In the assembly language of the PDP-10 computer, variable names could include the letters A–Z (just one set, not upper and lower case), the digits 0–9, and three punctuation characters (underscore, dollar sign, and percent sign). In addition, there was a NULL code for an unused character. What is the smallest number of bits that could be used to represent one character from this set? (Hint: $26 + 10 + 3 + 1 = 40$.)

Question 2 (1 point):

What is the smallest number of bits required to represent *two* characters from the set in question 1?

Question 3 (1 point):

In figure 1.1 of Patterson and Hennessy (page 7), we learn that the assembly language instruction

```
mulr $2,$5,5
```

is equivalent to the machine language instruction

```
00000000101000010000000000011000
```

How many bytes are required to represent the assembly language version? How many bytes are required to represent the machine language version?

Your name _____ login c60b-_____

Question 4 (4 points):

Write a C procedure named `down` that takes a character string as its argument. It should print a triangular display of the contents of the string, as shown in the following example: Invoking the procedure with the instruction

```
down("hello");
```

should print the five lines

```
h e l l o
h e l l
h e l
h e
h
```

(The number of lines printed is equal to the length of the string.) The procedure should not return anything. Use `putchar` to print each character individually. (Note the spaces printed between letters!)

Question 5 (4 points):

Write a MIPS assembler procedure equivalent to the following C procedure:

```
void setup(int a[], int len)
{
    int i;

    for (i=0; i<len; i++)
        a[i] = i;
}
```

You should assume that when your procedure is entered, register 4 contains the address of the array `a`, and register 5 contains the value `len`.