

Fall 2010 CS61C Midterm 1

Your Name: _____

Your TA: Andrew Michael Conor Charles

Login: cs61c-____

This exam is worth 85 points, or about 10% of your total course grade. The exam contains 7 questions.

This booklet contains 9 numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper.

Question	Points (Minutes)	Score
1	9 (8)	
2	8 (8)	
3	12 (8)	
4	9 (9)	
5	14 (12)	
6	9 (10)	
7	24 (30)	
Total	85 (85)	

Question 1 – Potpourri (9 points / 8 minutes)

a) Suppose we have defined the C structure:

```
struct player {
    int id;
    int numGoals;
    char name[8];
};
```

Also, we declare

```
struct player players[3];
```

such that `players` starts at `0x10000000`. What is the value of `playerTwo` after:

```
struct player *playerTwo = players + 2;
```

- a) Indeterminate: uninitialized memory
- b) `0x10000002`
- c) `0x10000020`
- d) `0x10000032`

b) Fill in the blank using one of the choices below. The quantity of numbers that single precision floating point can represent is _____ the quantity of numbers a 32-bit two's complement integer can represent.

- a) greater than
- b) less than
- c) the same as

c) True / False. Circle the right answer

I) T / F The most expensive memory per bit is the smallest memory in a memory hierarchy.

II) T / F The largest capacity memory is the slowest memory in a memory hierarchy.

III) T / F For a given instruction set architecture and hardware implementation, a compiler that produces fewer instructions always produces a faster program.

IV) T / F Data level parallelism is enabled by many small and independent tasks that can be spread among identical servers.

V) T / F In map-reduce processing, the reduction step must wait until it has received data from all of the mapping steps before it can start.

Question 2 Curves – (8 points / 8 minutes)

“Magic lives in curves, not angles.” Mason Cooley

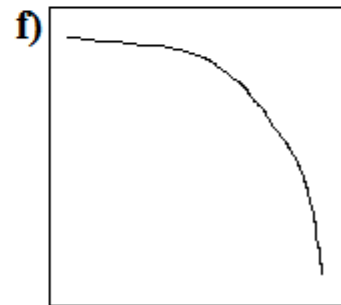
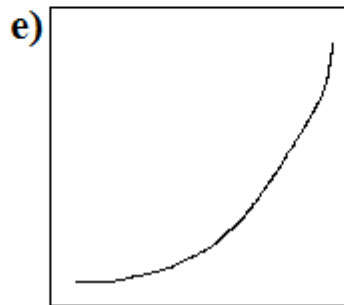
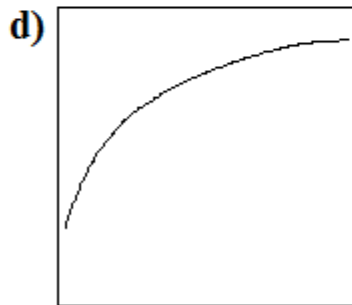
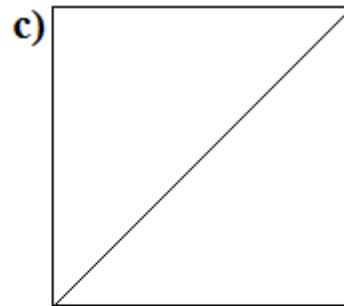
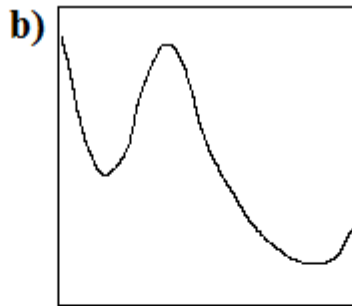
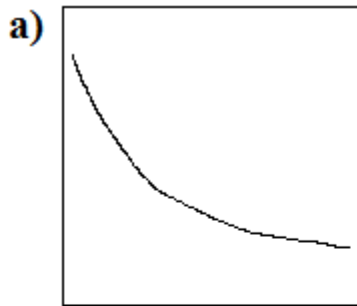
Please match each of the following descriptions with the graphs below. **Assume a linear scale for both the X and Y axes.**

I) A technology’s performance near the end of its lifespan. **(time on X Axis)** _____

II) Energy-proportional server energy usage as a function of utilization. **(utilization on X Axis)** _____

III) Density of transistors on a chip during the 90s. **(time on X Axis)** _____

IV) Typical server utilization histogram for servers in a datacenter. _____

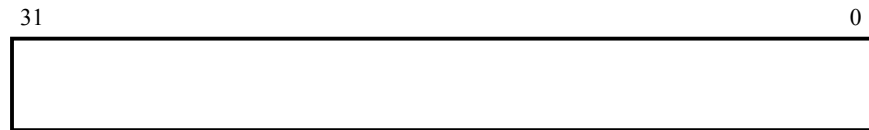


Question 3 – Hidden Treasure (12 points / 8 minutes)

“A box without hinges, key, or lid, yet golden treasure inside is hid.” J. R. R. Tolkien

- a) Consider a 32-bit byte address machine with a direct mapped cache. The cache is organized with 4096 blocks of four 32-bit words, each block using write back cache policy.

Draw and label the partitionings of the 32-bit memory address into the segments that are used to access the cache; Label each segment with its name and its width in bits. (show bit numbers)



- b) Including all of the cache management bits, what is the total number of bits in the cache?
Show your work, and to simplify the calculation, give your answer in kbits (1024 bits).

Question 4 – Shifty (9 points / 9 minutes)

“Twenty years of schoolin’ / And they put you on the day shift.” Bob Dylan

It was mentioned in lecture that computer designers had been fooled into thinking an arithmetic shift right (SRA) instruction is identical in effect to dividing by two.

SRA performs identically to shift right logical (SRL) for positive two’s complement numbers, but fills the leftmost bits with 1’s for negative numbers (most significant bit 1). Here is an example assuming an 8 bit word

SRA by 3 on 0010 1001 \Rightarrow 0000 0101.

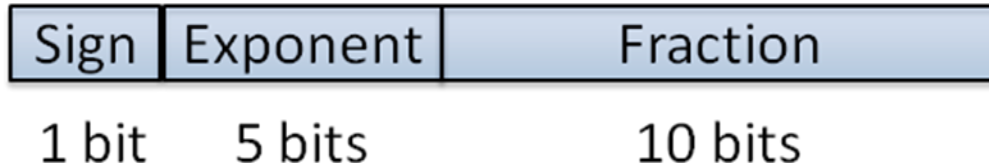
SRA by 3 on 1010 1001 \Rightarrow 1111 0101.

For one byte Two’s Complement integers, give an example that shows SRA cannot be used to implement the C *signed* integer division operator for powers of two. **You only need to use 8-bit words in your example.**

Question 5 – Halfway to Floating Point (14 points / 12 minutes)

“Anyone who thinks there's safety in numbers hasn't looked at the stock market pages.” Irene Peter

(Half Precision) The recent revision of the floating-point standard added 16-bit floating-point precision, with the following format:



- a) Assuming that half-precision follows the same philosophy as single and double precision, what should be the bias for this 5-bit exponent in half precision? _____
- b) What real *decimal* number (or symbol) do you expect each of the following binary bit patterns represent if they are half precision floating point numbers?

- I) 0 00000 0000000000 _____
- II) 0 01111 0000000000 _____
- III) 1 10000 0000000000 _____
- IV) 1 11111 0000000000 _____
- V) 1 11111 1111111111 _____

Question 6 – Returning Mystery (9 points / 10 minutes)

“Mystery is at the heart of creativity. That, and surprise.” Julia Cameron

What does the assembly function `mystery` return? Write your answer as a **binary number**. In **one short phrase**, describe how you got your answer. You may use some kind of obvious shorthand to denote long strings of ones or zeros, e.g. `16{1}` to denote 16 ones in a row.

Address		Instruction
0x08001000	<code>mystery:</code>	<code>addiu \$sp, \$sp, -4</code>
0x08001004		<code>sw \$ra, 0(\$sp)</code>
0x08001008		<code>addiu \$v0, \$zero, 0</code>
0x0800100c		<code>jal inner</code>
0x08001010		<code>lw \$ra, 0(\$sp)</code>
0x08001014		<code>addiu \$sp, \$sp, 4</code>
0x08001018		<code>jr \$ra</code>
0x0800101c	<code>inner:</code>	<code>lw \$v0, 4(\$ra)</code>
0x08001020		<code>jr \$ra</code>

Question 7 – Metasyntactic Function (24 points / 30 minutes)

“Garply: A metasyntactic variable like foo, once popular among SAIL hackers.”

Computing Dictionary

a) Examine the following MIPS function `garply`. Some of the instructions related to function calling conventions have been omitted. Fill in the instructions so that `garply` can be used correctly. Then assemble the 6 instructions that were given in the problem into binary MIPS machine code. Give your answers in binary, and then hexadecimal. Assume that the base address of the function `garply` is at `0x10000000`. You may use some kind of obvious shorthand to denote long strings of ones or zeros, e.g. `16{1}` to denote 16 ones in a row. See examples below.

MIPS assembly	Machine Code	
	Binary	Hexadecimal
<code>garply:</code>		

<code>addiu \$v0 \$zero 0</code>	<code>001000 00000 00010 16{0}</code>	<code>0x20020000</code>
<code>lbu \$t0 0(\$a0)</code>		
<code>beq \$t0 \$zero end</code>		

<code>addiu \$a0 \$a0 1</code>		

<code>jal garply</code>		

<code>addiu \$v0 \$v0 1</code>	<code>001000 00010 00010 (15{0}) 1</code>	<code>0x20420001</code>
<code>end:</code>		

b) Give a direct translation of `garply` into C. Don't change how the function computes its answer (i.e., don't change recursion into iteration, or vice-versa, or anything like that). Your solution shouldn't take more than 3 lines of code, but we'll give you up to six. Don't forget to correctly fill in the argument and return types!

```

____ garply(____) {
_____
_____
_____
_____
_____
_____
_____
}

```

} (Question cont'd on next page)

Question 7 – (Cont'd)

c) What is the behavior of `garply`? If you can think of a standard library function with the same behavior, feel free to just state its name. If you can't think of such a library function, give a *concise* English description of the behavior of `garply`. Use no more than a single sentence.