

Full Name: \_\_\_\_\_

'On my honor as a University  
of Colorado at Boulder student  
I have neither given nor received  
unauthorized assistance on this  
work.'

## CSCI 2400, Fall 2013

### Sample Problems for Midterm Exam 1

#### Instructions:

- Make sure that your exam is not missing any sheets, then write your full name on the front. Put your name or student ID on each page.
- Write your answers in the space provided below the problem. If you make a mess, clearly indicate your final answer.
- Feel free to use the back of pages to work out answers. Indicate when you have done so with an arrow or "BACK".
- This exam is CLOSED BOOK and you can use a *single page* of notes along with our reference sheets. Please attach your single sheet of notes to your exam when you're done. You can not use a computer or calculator. Good luck!

Problem	Page	Possible	Score
	<b>Total</b>	<b>0</b>	

1. Assume we are running code on a 8-bit machine using two's complement arithmetic for signed integers. Also assume that  $T_{Max}$  is the maximum integer,  $T_{Min}$  is the minimum integer. Fill in the empty boxes in the table below. The following definitions are used in the table:

```
int y = -12
int x = 22;
```

Note: You need not fill in entries marked with “–”.

Each blank space is 1 point.

In the column labeled “Over/Under”, you should indicate if an overflow (carry out of the highest bit) or underflow (borrow from the highest bit) occurred.

Expression	Decimal Representation	Hex Representation	Over/Under?
–	31	0x1f	–
–	-3	0xfd	–
y	-12	0xf4	–
x+y	10	0xA	No
x + $T_{Max}$	-107	0x95	Yes
$T_{Min}$ -x	106	0x6a	Yes

2. Despite the budget sequestration, the USA decides to send a person to the moon. But, due to the budget cuts, they can only afford a computer that has 12 bit floating point numbers *bf with truncation mode*. This includes a sign bit, 6-bit exponent field with bias 31 and a 5 bit mantissa / fractional field).

- (a) Show what 1 011001 01010 represents in this floating point format? Convert the mantissa portion to a decimal (base 10) number (*e.g.* 1.234) and show the exponent expressed as a power of two. It may be useful to know that  $1/2 = 0.5$ ,  $1/4 = 0.25$ ,  $1/8 = 0.125$ ,  $1/16 = 0.0625$ ,  $1/32 = 0.03125$ . Your resulting number should be formatted something like  $1.234 * 2^{56}$ , but you should obviously write out the proper value represented by 1 011001 01010.

\_\_\_\_\_ . \_\_\_\_\_ \* 2<sup>-----</sup>

**Answer:**

$$\begin{aligned}
 &= -(1 + (\frac{10}{32})) \times 2^{-25-31} \\
 &= -(1 + 0.25 + 0.0625) \times 2^{-6} \\
 &= -1.3125 \times 2^{-6}
 \end{aligned}$$

- (b) The moon moves away from the earth approximately  $\frac{4}{1024}$  km per year<sup>1</sup> How many mantissa bits would be needed to add together the diameter of the moon plus the distance the moon moves away from the earth in one year. In otherwords, how many bits are needed in the fractional part to *precisely* represent  $1737 + \frac{4}{1024}$ .

**Answer:**

You need 11 bits to represent the original number (1.1011001001). Then  $\frac{4}{1024} = \frac{1}{256}$  km, and need 8 bits to represent that, for a total of  $11 + 8 = 19$  bits, but can use “implicit 1” to get 18 bit fractional part.

<sup>1</sup>It does. Look it up.

## 3. Given the following code

Given this C code	is translated to
<pre> struct node {     int x;     struct node *left, *right; };  int sum(struct node *p) {     if ( p ) {         return sum( p -&gt; left )             + sum( p -&gt; right )             + p-&gt;x;     } else {         return 0;     } } </pre>	<pre> sum:     pushl    %ebp     movl     %esp, %ebp     subl     \$12, %esp     movl     %ebx, -8(%ebp)     movl     %esi, -4(%ebp)     movl     8(%ebp), %ebx     movl     \$0, %eax     testl    %ebx, %ebx     je       .L4      movl     4(%ebx), %eax     movl     %eax, (%esp)     call     sum     movl     %eax, %esi     movl     8(%ebx), %eax     movl     %eax, (%esp)     call     sum     leal     (%esi,%eax), %eax     addl     (%ebx), %eax  .L4:     movl     -8(%ebp), %ebx     movl     -4(%ebp), %esi     movl     %ebp, %esp     popl     %ebp     ret </pre>

Draw a picture of the stack frame as it would be at the point of the “je” instruction in the program, including any arguments passed to the function. Label each word in the stack frame and explain what each word is used for. If a word is used for multiple things, mention both. If it is not used, mention that.

	p	argument (“p”) from caller
	R/A	return address from caller
BEP →	oldbep	Saved BEP from caller
	oldesi	Saved ESI from caller
	oldebpx	Saved EBX from caller
ESP →	arg	Space for argument for recursive calls to “sum”