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 2014
First Midterm Exam

Instructions:

- Check that your exam has all 4 pages, and write your full name clearly on the front.
- Write your answers in the space provided for each problem. Feel free to use the back of each page to help you determine the answer, but make sure your answer is entered in the space provided on the front of the page.
- This exam is CLOSED BOOK and no electronics are allowed. You can use one page of personal notes and the printed midterm packet of tables. Good luck!

Problem	Page	Possible	Score
1	1	8	
2	2	16	
3	2	18	
4	3	28	
5	4	30	
Total		100	

1. **[8 Points]** In the following, state whether the statement is true or false. An incorrect answer will cancel a correct answer. The lowest possible score is zero on this question.

   (a) _____ In big endian systems, the most significant byte of a word has the lowest memory address.

   (b) _____ omitted

   (c) _____ Alignment rules require a float to be aligned on a memory address that is a multiple of 8.

   (d) _____ 64-bit x86 assembly does not employ a separate register for a frame pointer.

2. **[16 Points]** In the following questions assume the variable `x` is a signed integer and that the machine uses two's complement representation. Also assume that `TMax` is the maximum integer, `TMin` is the minimum integer, and `W` is one less than the word length (e.g., `W` = 31 for 32-bit integers).

   Match each of the descriptions on the left with a line of code on the right (write in the letter in the blank). You will be given 5 points for each correct match.

   ```
   1) x == 1    _____        a) (x | (~x)) ^ 0

   2) -1        _____        b) ! (x ^ 0)

                             c) ~TMin + 1

                             d) !(x ^ 1)
   ```

3. **[18 Points]** Assume we are running code on a 5-bit machine using two's complement arithmetic for signed integers. Also assume that `TMax` is the maximum integer, `TMin` is the minimum integer. Fill in the empty boxes in the table below. The following definitions are used in the table:

   ```
   int x = 10;
   int y = -7;
   ```

   Note: For the empty boxes in the first column (Expression) you MUST USE either x or y along with any other constants, e.g. x+17. In the column labeled "Overflow", you should indicate "Yes" or "No" whether overflow occurred. The overflow could be either on the positive or negative side. Each blank is worth 2 points.

Expression	Decimal Representation	Hex Representation	Overflow?
x	10		No
		0x11	Yes
y-9	TMin		No
x+y			No
y + TMax			
y + TMin	9	0x09	

4. **[28 Points]** omitted

5. **[30 Points]**

Look at the C code below and pick the correct option to fill out the blanks in the corresponding assembly code. Options for blanks: `%eax, %ebx, %ecx, %edx, %esi, %edi, %rsp, %rbp, add, sub, imul, jmp, je, jne, js, jle, jge, jl, jg, cmpl, lea, mov`. Options may be used more than once. Each blank is worth 4 points.

C Code:

```
int mysteryFunction (int x,char c)
{
  signed int result = 0;
  unsigned short int i;
  char b[5];

  for (i=1; i<=x; i++){
    if(c -'a' >= 0){
      result = result + i*i;
    }
    else {
      result = result + i;
    }
    b[i-1] = result + 'a';
  }

  return result;
}
```

```
mysteryFunction:
push   %rbp
mov    %rsp,%rbp
mov    %edi,-0x14(%rbp)
mov    %esi,%eax
mov    %al,-0x18(%rbp)
movl   $0x0,-0x4(%rbp)
movw   $0x1,-0x6(%rbp)
_____  L5

.L2
movsbl -0x18(%rbp),%eax
_____   $0x61,%eax
test    %eax,%eax
_____   L3
movzwl -0x6(%rbp),%edx
movzwl -0x6(%rbp),%eax
_____  %edx,%eax
add    %eax,-0x4(%rbp)
jmp    L4

.L3
movzwl -0x6(%rbp),%eax
add    %eax,-0x4(_______)
jmp    L4

.L4
movzwl -0x6(%rbp),%eax
_____  -0x1(%rax),%ecx
mov    -0x4(%rbp),%eax
add    $0x61,_______
mov    %eax,%edx
movslq %ecx,%rax
mov    %dl,-0x10(_______,%rax,1)
movzwl -0x6(%rbp),%eax
_____  $0x1,%eax
mov    %ax,-0x6(%rbp)
jmp    L5

.L5
movzwl -0x6(%rbp),%eax
cmp    -0x14(%rbp),%eax
_____   L2
mov    -0x4(%rbp),%eax
pop     %rbp
retq
```