ECE 198JL First Midterm Exam
Spring 2013

Tuesday, February 5th, 2013

Name: ______________________ NetID: ____________

Discussion Section:

10:00 AM	[ ] JD1	
11:00 AM	[ ] JD2	
2:00 PM	[ ] JD3	
4:00 PM	[ ] JD4	

- **Be sure your exam booklet has 9 pages.**
- **Be sure to write your name and lab section on each exam page.**
- **Do not tear the exam booklet apart; you can only detach the last page.**
- **We have provided an ASCII table at the back.**
- **Use backs of pages for scratch work if needed.**
- **This is a closed book exam. You may not use a calculator.**
- **You are allowed one handwritten 8.5 x 11" sheet of notes.**
- **Absolutely no interaction between students is allowed.**
- **Be sure to clearly indicate any assumptions that you make.**
- **The questions are not weighted equally. Budget your time accordingly.**
- **Don't panic, and good luck!**

Problem 1	20 points:	________
Problem 2	20 points:	________
Problem 3	15 points:	________
Problem 4	25 points:	________
Problem 5	20 points:	________
Total	100 points:	________

 Name: ______________________

Problem 1 (20 pts): Binary Representation

1. Convert the following decimal numbers to 8-bit 2's complement binary representation. Show your work.

a) 63_{10} = $0011\ 1111_2$

b) -94_{10} = $1010\ 0010_2$

	Odd?	
63	1	$\rightarrow (63-1)/2 = 31$
31	1	$\rightarrow (31-1)/2 = 15$
15	1	$\rightarrow (15-1)/2 = 7$
7	1	$\rightarrow (7-1)/2 = 3$
3	1	$\rightarrow (3-1)/2 = 1$
1	1	$\rightarrow$ Done

	Odd?	
94	0	$\rightarrow 94/2 = 47$
47	1	$\rightarrow (47-1)/2 = 23$
23	1	$\rightarrow (23-1)/2 = 11$
11	1	$\rightarrow (11-1)/2 = 5$
5	1	$\rightarrow (5-1)/2 = 2$
2	0	$\rightarrow 2/2 = 1$
1	1	$\rightarrow$ Done

$94_{10} = 0101\ 1110_2$

$1010\ 0001 + 1$

$10100010_2 = -94_{10}$

2. Convert the following 2's complement binary number to decimal. Show your work.

Negative

$\hookrightarrow 10010010_2 = -110_{10}$

$01101101 + 1$

$01101110_2 = 2^6 + 2^5 + 2^3 + 2^2 + 2^1 = 64 + 32 + 8 + 4 + 2 = 110_{10}$

3. Convert the following decimal number to 32-bit floating point binary representation. Show your work.

-2.0625 = $1\ 1000\ 0000\ \ 0000\ 1000\ 0000\ 0000\ \ 0000\ 000$

$\downarrow$

$S = 1$

$0.0625 \times 2 = 0.1250 \rightarrow 0$

$0.1250 \times 2 = 0.2500 \rightarrow 0$

$0.2500 \times 2 = 0.5000 \rightarrow 0$

$0.5000 \times 2 = 1.0000 \rightarrow 1$

$1.00001_2 \times 2^1_{10}$

$\hookrightarrow$ Fraction $= 0000\ 1000\ 0000\ 0000\ 0000\ 000$

$\hookrightarrow$ Exponent $= 1_{10} + 127_{10} = 128_{10} = 1000\ 0000_2$ (Unsigned)

$2.0625_{10} = 10.0001_2 = 1.00001_2 \times 2^1_{10}$ (Unsigned)

4. Convert the following 32-bit binary pattern to the corresponding decimal representation of IEEE 574 floating point number. Show your work.

S / Exponent / Fraction

0 10000100 00000010000000000000000 = 32.25

$2^7 + 2^2 = 128 + 4 = 132$

$1.0000\ 0010_2 = 1 + 2^{-7}_{10}$

$$N = (-1)^S \times 1.\text{Fraction} \times 2^{\text{Exp.} - 127}$$

$$= (-1)^0 \times 1.0000\ 0010_2 \times 2^{132-127}_{10} = (1 + 2^{-7}_{10}) \times 2^5_{10} = 2^5_{10} + 2^{-2}_{10} = 32 + \frac{1}{4} = 32 + 0.25 = 32.25$$

Problem 2 (20 pts): Operations on Binary Numbers

1. Compute the following arithmetic operations on eight-bit 2's complement numbers. Express your answer as an 8-bit 2's complement number. Indicate if it has an overflow by circling the corresponding YES or NO.

a) 00110011 + 00010100 = 0100 0111 Overflow? YES NO

0011 0011+
0001 0100
0 1 0 0 0 1 1 1

b) 01101010 - 10111001 = 1011 0001 Overflow? YES NO

01000110+
1
0100 0111

0110 1010+
0100 0111
1 011 0001

c) 10110101 + 10111100 = 0111 0001 Overflow? YES NO

1011 0101+
1011 1100
01110001

d) 10111111 - 10111100 = 0000 0011 Overflow? YES NO

01000011+
1
01000100

1011 1111+
0100 0100
0000 0011

e) 01010100 + 00101010 = 0111 1110 Overflow? YES NO

0101 0100+
00101010
01111110

2. For the three eight-bit binary numbers, A = 101, B = 110, and C = 011, give the result of the following bitwise logical operations.

a) (A AND B) OR C = 111

101 AND
110
100

100 OR
011
111

b) (A OR B) AND C = 011

101 OR
110
111

111 AND
011
011

3. When subtracting 0.3 from 100.0 using 32-bit floating point representation, the result is 99.699997 (x42c76666) instead of 99.7. Explain why this happens.

With 32-bit floating point representation we cannot represent the number 99.7 exactly, we need to round its representation to the available number of bits for the fraction (23 bits).

 Name: ______________________

Problem 3 (15 pts): Codes, Error Detection and Correction

1. If a code has distance d = 5, then it can detect up to how many errors?

   Answer: $d-1=4$

2. If a code has distance d = 6 then it can correct up to how many errors?

   Answer: $\left\lfloor \frac{d-1}{2} \right\rfloor = \lfloor 2.5 \rfloor = 2$

3. Specify the distance of each of the following codes

   2 2 2

   a) {000, 011, 101, 110}   distance = 2

   2 2 2

   b) 7-bit ASCII code   distance = 1

4. For $\boxed{\text{7-bit}}$ ASCII characters 'B' and 'C' write in binary their corresponding 8-bit $\boxed{\text{odd}}$ parity representations. (Put parity bit in front of 7-bit ASCII representation.)

   'B' = 42_{16} = $100\ 0010_2$

   a) 'B' = 1 100 0010

   b) 'C' = 0 100 0011   'C' = 43_{16} = $100\ 0011_2$

5. For $\boxed{\text{7-bit}}$ ASCII characters ~~'%'~~ '&' and '3' write in binary their corresponding 8-bit $\boxed{\text{even}}$ parity representations. (Put parity bit in front of 7-bit ASCII representation.)

   a) '&' = 1 010 0110   '&' = 26_{16} = $010\ 0110_2$

   b) '3' = 0 011 0011   '3' = 33_{16} = $011\ 0011_2$

First 6 user entries:
Smallest #: -7, 2nd smallest #: -4

Problem 4 (25 pts): C program analysis

Consider the following "mystery" C program, to which the inputs 5, 44, 67, -4, 11, -7, 60, 55 will be given until the program terminates. (Note that the program may not scan all of those values.) For this problem, analyze and execute the program in your head (you can make notes on this page or on the scratch pages if needed) to find the results of the computation.

```
/* mystery.c */

#include <stdio.h>

#define A_VAL 6
#define MAX_VAL 9999

int main()
{
      int ii;
      int value;
      int value1 = MAX_VAL;
      int value2 = MAX_VAL;

      for ( ii = 0; ii < A_VAL; ii = ii + 1 ) → This loop will execute 6 times
      {
          scanf("%d", &value); → Program will get first 6 user entries and
          if ( value < value1 )    store them in value sequentially
          {
              value2 = value1;  If this compound statement executes in a
              value1 = value;   given iteration we will have:
          }                     value = value1 < value2
          else
          if ( value < value2 )
          {
              value2 = value;  If this statement executes in a given iteration
          }                    we will have value1 ≤ value2 = value

          /* CHECKPOINT FOR PART 1 */
      } → Thus, loop gets first 6 user entries and stores smallest entry in value1, second smallest
          entry in value2
      printf("The output value is %d\nGoodbye!", value2);
                                                   ↓
      return 0;                                   -4
}
```

 Name: ______________________________

1. At the location in the program marked "CHECKPOINT FOR PART 1", determine and list the current values of the variables for each time that the program reaches that checkpoint. Fill in only as many rows as needed below.

ii =	0	value =	5	value1 =	5	value2 =	9999
ii =	1	value =	44	value1 =	5	value2 =	44
ii =	2	value =	67	value1 =	5	value2 =	44
ii =	3	value =	-4	value1 =	-4	value2 =	5
ii =	4	value =	11	value1 =	-4	value2 =	5
ii =	5	value =	-7	value1 =	-7	value2 =	-4
ii =		value =		value1 =		value2 =	
ii =		value =		value1 =		value2 =	

2. Write down EXACTLY the formatted text that will be printed on the terminal screen by the final `printf` statement in the program.

The output value is -4
Goodbye!

3. Complete the following sentence to describe the computational task performed by this "mystery" program.

The "mystery.c" program finds the two smallest numbers

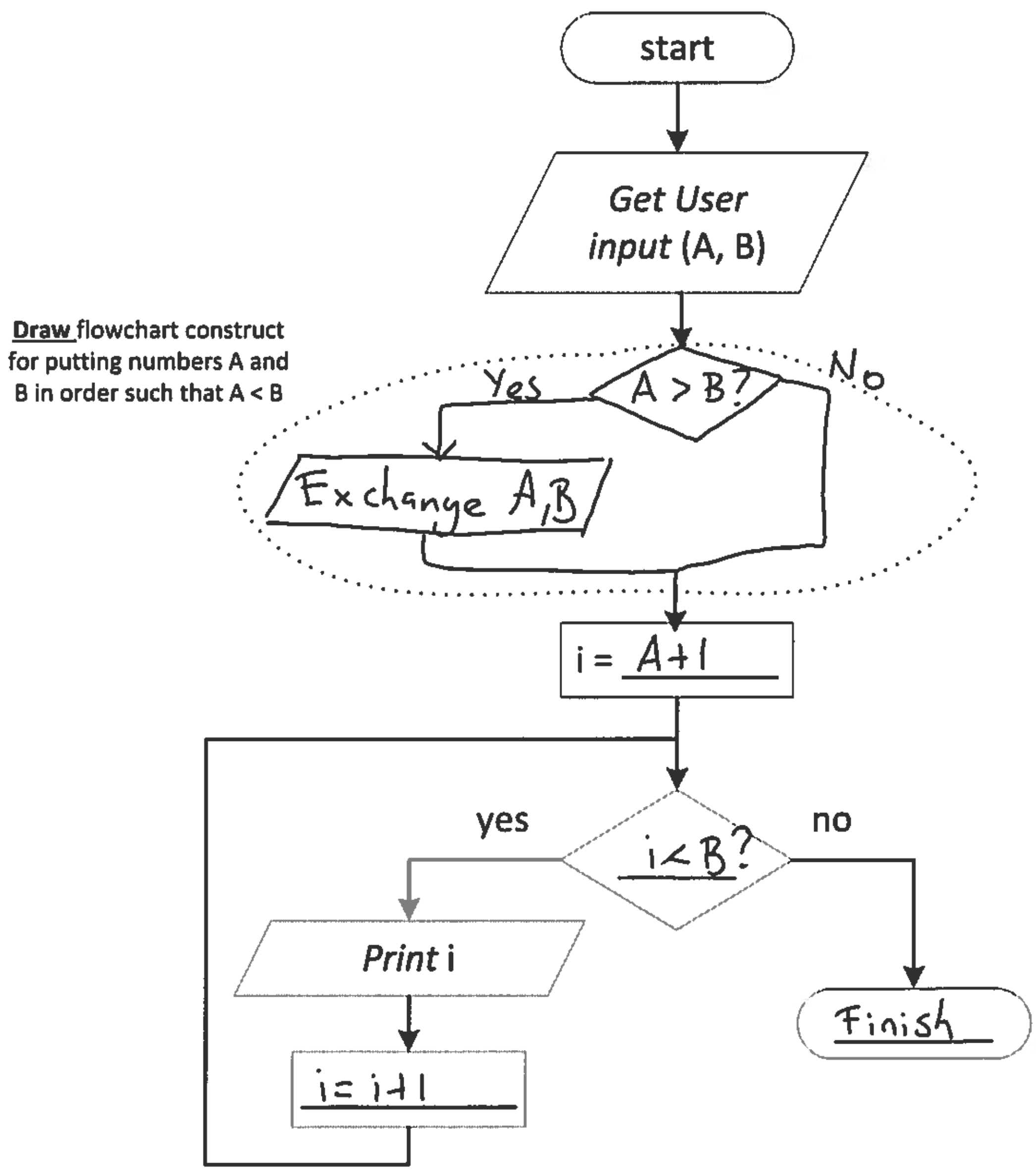
of a series of six [tell how many] integer input values.

 Name: ___________________________

Problem 5 (20 pts): Programming in C

Write a program that asks user to enter two integer numbers and outputs the sequence of integer numbers in between. For example, for inputs 2 and 5, the numbers in between are 3 and 4. *Note that user inputs may not be ordered.* That is, user can either enter 2 and 5, or 5 and 2 as two input numbers, and the program still should produce the correct sequence of numbers: 3 and 4.

1. Fill in the missing parts in the flowchart below.



2. Fill in missing lines of code to complete the program.

```
#include <stdio.h>

int main()
{
    int  A, B;
    int tmp, i;

    printf("Enter two integer numbers: ");
    scanf("%d %d", &A, &B);

    /* put the inputs in the correct order, A<B */
    if (A > B)
    {
        tmp = A;
        A = B;
        B = tmp;
    }

    /* output the numbers in between A and B */
    for (i = A+1; i < B; i = i+1)
    {
        printf("%d\n", i);
    }

    return 0;
}
```

 Name: ______________________________

Table of ASCII Characters

Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex
(nul)	0	00	(sp)	32	20	@	64	40	`	96	60
(soh)	1	01	!	33	21	A	65	41	a	97	61
(stx)	2	02	"	34	22	B	66	42	b	98	62
(etx)	3	03	#	35	23	C	67	43	c	99	63
(eot)	4	04	$	36	24	D	68	44	d	100	64
(enq)	5	05	%	37	25	E	69	45	e	101	65
(ack)	6	06	&	38	26	F	70	46	f	102	66
(bel)	7	07	'	39	27	G	71	47	g	103	67
(bs)	8	08	(	40	28	H	72	48	h	104	68
(ht)	9	09	)	41	29	I	73	49	i	105	69
(nl)	10	0a	*	42	2a	J	74	4a	j	106	6a
(vt)	11	0b	+	43	2b	K	75	4b	k	107	6b
(np)	12	0c	,	44	2c	L	76	4c	l	108	6c
(cr)	13	0d	-	45	2d	M	77	4d	m	109	6d
(so)	14	0e	.	46	2e	N	78	4e	n	110	6e
(si)	15	0f	/	47	2f	O	79	4f	o	111	6f
(dle)	16	10	0	48	30	P	80	50	p	112	70
(dc1)	17	11	1	49	31	Q	81	51	q	113	71
(dc2)	18	12	2	50	32	R	82	52	r	114	72
(dc3)	19	13	3	51	33	S	83	53	s	115	73
(dc4)	20	14	4	52	34	T	84	54	t	116	74
(nak)	21	15	5	53	35	U	85	55	u	117	75
(syn)	22	16	6	54	36	V	86	56	v	118	76
(etb)	23	17	7	55	37	W	87	57	w	119	77
(can)	24	18	8	56	38	X	88	58	x	120	78
(em)	25	19	9	57	39	Y	89	59	y	121	79
(sub)	26	1a	:	58	3a	Z	90	5a	z	122	7a
(esc)	27	1b	;	59	3b	[	91	5b	{	123	7b
(fs)	28	1c	<	60	3c	\	92	5c	|	124	7c
(gs)	29	1d	=	61	3d	]	93	5d	}	125	7d
(rs)	30	1e	>	62	3e	^	94	5e	~	126	7e
(us)	31	1f	?	63	3f	_	95	5f	(del)	127	7f