ECE 120 First Midterm Exam
Fall 2016

Tuesday, September 20, 2016

Name: SOLUTIONS

NetID: ______________

Discussion Section:

9:00 AM		
10:00 AM		
11:00 AM	[ ] AB1	[ ] AB8
12:00 PM	[ ] AB2	[ ] AB9
1:00 PM	[ ] AB3	[ ] ABA
2:00 PM	[ ] AB4	[ ] ABB
3:00 PM	[ ] AB5	
4:00 PM	[ ] AB6	[ ] ABC
5:00 PM	[ ] AB7	[ ] ABD

- **Be sure that your exam booklet has 12 pages.**
- **Write your name, netid and check discussion section on the title page.**
- **Do not tear the exam booklet apart.**
- **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 (both sides).**
- **Absolutely no interaction between students is allowed.**
- **Clearly indicate any assumptions that you make.**
- **The questions are not weighted equally. Budget your time accordingly.**
- **Show your work.**

Problem 1	14 points	________
Problem 2	18 points	________
Problem 3	8 points	________
Problem 4	18 points	________
Problem 5	18 points	________
Problem 6	24 points	________
Total	100 points	________

Problem 1 (14 points): Number systems

1. **(4 points)** Consider the following 16 bits: **0011 1010 0010 1001.** Give the hexadecimal representation of these bits and interpret them as a string of 8-bit ASCII characters. Use the ASCII table on the last page of the exam.

   Hexadecimal: 3A29

   ASCII characters: :)

2. **(2 points)** There are 632 pages in *Introduction to Computing Systems (2nd edition)* by Yale Patt and Sanjay Patel. If the authors decided to number the pages using fixed-length binary words, what is the minimum number of bits they would use per page number?

   Minimum number of bits per page number: 10 (decimal number)

3. **(4 points)** Convert the decimal number **222** to 8-bit unsigned: 11011110

4. **(4 points)** Convert the 8-bit 2's complement number **1010 1010** to decimal: -86

Problem 2 (18 points): 2's complement arithmetic

1. **(9 points)** As the barista pours you a drink in the Daily Byte Café, you wonder what the ECE building would be like without its café. Treating **xECEB** and **xCAFE** as 16-bit 2's complement numbers, calculate the value of **xECEB − xCAFE** and write it as a 16-bit 2's complement number in hexadecimal representation. Does overflow occur?

   xECEB − xCAFE = x21ED (2's complement number in hexadecimal representation)

   Circle one: **OVERFLOW** [✓ **NO OVERFLOW**]

2. **(9 points)** Now gazing out into the ECE building's atrium with drink in hand, you begin to daydream about adding a Fine & Applied Arts (FAA) minor to your ECE major. Treating **xECE** and **xFAA** as 12-bit 2's complement numbers, calculate the value of **xECE + xFAA** and write it as a 12-bit 2's complement number in hexadecimal representation. Does overflow occur?

   xECE + xFAA = xE78 (2's complement number in hexadecimal representation)

   Circle one: **OVERFLOW** [✓ **NO OVERFLOW**]

Problem 3 (8 points): Logical operations

1. **(4 points)** Perform the following bitwise logical operations. Express each answer as a single hexadecimal digit.

   a) xE XOR x4 = xA (answer in hexadecimal representation)

   b) NOT(xC OR (NOT(x5))) = x1 (answer in hexadecimal representation)

2. **(4 points)** Fill in the truth table for the following Boolean expression:

a	b	c	a AND (NOT(b OR c))
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Problem 4 (18 points): Floating-point representation

1. **(6 points)** What decimal number is represented by the bit pattern below in IEEE single-precision floating-point format? **Show your work.**

FIRST GROUP OF 16 BITS

1	1	0	0	0	0	0	1	0	0	1	1	0	1	0	0

zeroes

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SECOND GROUP OF 16 BITS

→ negative

$2^7+2^1=130$

$130-127=3$

so 2^3

$-\underset{\text{implicit}}{1}.\ 01101 \times 2^3$

$-1.01101 \times 2^3 = -1011.01 = \boxed{-11.25}$

2. **(6 points)** How many different **normalized** numbers can be represented with the IEEE 754 single-precision floating-point format? Use the IEEE 754 format on the last page of the exam.

2x for sign bit

2^{23}x for mantissa

254 choices for exponent

$2 \times 2^{23} \times 254 = \boxed{2^{24} \times 254}$

Problem 4 (18 points), continued:

3. **(6 points)** The number below is represented with IEEE single-precision floating-point format. Multiply the number by 128 and write the result in the boxes below, again using the IEEE single-precision floating-point representation.

The number to be multiplied:

FIRST GROUP OF 16 BITS

0	0	1	1	1	0	0	0	0	1	0	1	0	1	0	1

0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0

SECOND GROUP OF 16 BITS

copy

add 7

copy

Write your answer in these boxes:

FIRST GROUP OF 16 BITS

0	0	1	1	1	0	1	1	1	1	0	1	0	1	0	1

0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0

SECOND GROUP OF 16 BITS

$128 = 2^7 \Rightarrow$ add 7 to exponent

$7 = 111$ in base 2

Problem 5 (18 points): C Program Analysis

Consider the following C program, which is provided inputs 5, 2, 4, 6, 8, 10, 12 until the program terminates. Trace the program to find the results of the computation performed (make notes on this page or on the scratch pages if needed). Answer the questions on the next page.

```
#include <stdio.h>

int main()
{
     int s = 100, num, i, k, val;

     scanf("%d", &num);


     for(i=1; i<num; i=i+1)
     {
          scanf("%d",&k);

          s = s - k;

          /* CHECKPOINT A */
     }
     val = (num*(num+1))/2;

     /* CHECKPOINT B */

     printf("The number is %d!!", val-s);

     return 0;
}
```

Problem 5 (18 points), continued:

(Inputs replicated from previous page for your convenience.) Assume that the numbers entered by the user are 5, 2, 4, 6, 8, 10, 12.

1. **(9 points)** At the location in the program marked "`CHECKPOINT A`," 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.

i =	1	k =	2	s =	98
i =	2	k =	4	s =	94
i =	3	k =	6	s =	88
i =	4	k =	8	s =	80
i =		k =		s =	
i =		k =		s =	
i =		k =		s =	

Out of for loop ~~Program terminates~~ So no entries

2. **(6 points)** Write down the values of the variables when the program reaches `CHECKPOINT B`.

`num =` 5 `val =` 15

3. **(3 points)** Write down the formatted text EXACTLY as will be printed on the screen when the final `printf` statement is executed.

The number is -65!!

Problem 6 (24 points): Finding the Exam Average with C

Prof. Lumetta wants to know the average exam score for the class, but spreadsheets intimidate him. Instead, he wants you to write a C program that lets him type in the scores for all 416 ECE120 students one by one, then prints out the average for him. Prof. Lumetta has started the program below, but he needs your help to finish it.

```
#include <stdio.h>

int main ()
{

    /* IN PART F, YOU MUST DECIDE ON VARIABLE INITIALIZATION. */

    float  sum;      /* sum of all students' scores   */
    int    student;  /* student number from 1 to 416 */
    float  score;    /* one student's exam score     */

    for (          /* YOUR ANSWER TO PART A WILL GO HERE. */ )
    {

                   /* YOUR ANSWER TO PART B WILL GO HERE. */

                   /* YOUR ANSWER TO PART C WILL GO HERE. */

                   /* YOUR ANSWER TO PART D WILL GO HERE. */
    }

                   /* YOUR ANSWER TO PART E WILL GO HERE. */

    /* Hurrah! */

    return 0;
}
```

Problem 6 (24 points), continued:

For each of the parts below, write C code in the boxes. Prof. Lumetta will copy your code from each box into the program at the position marked for that part of the question (A through F).

A. (6 points) For anonymity reasons, students must be called "1" through "416" in the program. Prof. Lumetta has declared the variable `student` to hold a student's number. Your first task is to fill in the `for` loop to make the loop body execute 416 times such that the value of the variable `student` runs from 1 to 416. Write your answer in the box below.

```
student=1 ; 416 >= student; student = student + 1
```

B. (4 points) Next, Prof. Lumetta needs the program to prompt him for a student's score (by number). For example, for student #42, the prompt should read

```
Student 42's score:
```

Write C code in the box below to produce the appropriate prompt for Prof. Lumetta. (*The prompt need not include spaces nor newlines/linefeeds before nor after the text shown.*)

```
printf("Student %d's score:  ", student);
```

C. (4 points) The program should then allow Prof. Lumetta to type in a real number representing a student's score. The score should be stored in the variable `score`. The format specifier for reading a `float` is `%f`. Write a C expression in the box below to allow Prof. Lumetta to type in one score.

```
scanf ("%f", &score);
```

Problem 6 (24 points), continued:

Prof. Lumetta wants to know the arithmetic average of the scores, but he only knows how to compute it mathematically by adding up the 416 scores and then dividing by 416. Write C code into the boxes below to compute and report the average that he wants. If the average is 87.557692, for example, the output should be "`The average is 87.557692.`" followed by an ASCII newline/linefeed. The format specifier to use is `%f`.

D. (3 points)

```
Sum= sum+score;
```

E. (4 points)

```
printf("The average is %f.\n", sum/416);
```

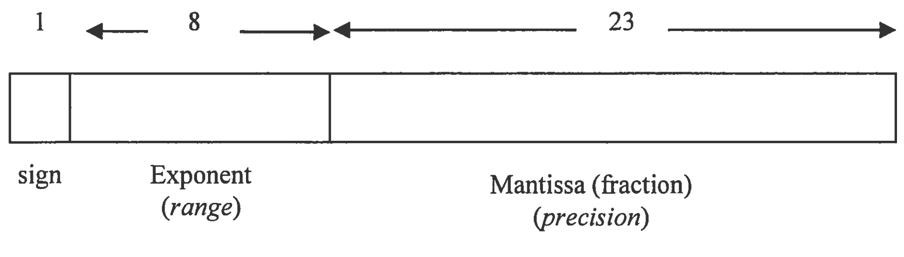
F. (3 points) Finally, for each of the variables in the program, fill in the box below with the value to which the variable should be initialized in the variable declarations at the top of `main`. If the variable need not be initialized, put an "X" in the box.

`sum`	`student`	`score`
0	X	X

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
(lf)	10	0a	*	42	2a	J	74	4a	j	106	6a
(vt)	11	0b	+	43	2b	K	75	4b	k	107	6b
(ff)	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

IEEE 754 32-bit floating point format



The actual number represented in this format is:

$$(-1)^{\boxed{s}} \times 1.\boxed{\text{mantissa}} \times 2^{\boxed{\text{exp.}} - 127}$$

where $1 \leq \text{exponent} \leq 254$ for normalized representation.