ECE 120 First Midterm Exam
Spring 2017

Tuesday, February 14, 2017

Name: ______________________________ NetID: ________________

Discussion Section and TA:

9:00 AM	[ ] AB1 Rui	
10:00 AM	[ ] AB2 Rui	
11:00 AM	[ ] AB3 Matt	
12:00 PM	[ ] AB4 Pawel	
1:00 PM	[ ] AB5 Pawel	
2:00 PM	[ ] AB6 Gowthami	[ ] ABA Huiren
3:00 PM	[ ] AB7 Gowthami	[ ] ABB Huiren
4:00 PM	[ ] AB8 Yu-Hsuan	[ ] ABC Sifan
5:00 PM	[ ] AB9 Yu-Hsuan	[ ] ABD Surya

- **Be sure that your exam booklet has 11 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	18 points	____________
Problem 2	16 points	____________
Problem 3	18 points	____________
Problem 4	15 points	____________
Problem 5	16 points	____________
Problem 6	17 points	____________
Total	100 points	____________

Problem 1 (18 points): Representations

1. For each of the following representations, **extend** the bit pattern **101001 from 6 bits to 8 bits**. The number represented should not change.

    a. **(2 points)** 6-bit unsigned to 8-bit unsigned: __________________

    b. **(2 points)** 6-bit 2's complement to 8-bit 2's complement: __________________

2. Encode the **decimal value 88** into each of the following representations:

    a. **(2 points)** 8-bit unsigned: __________________

    b. **(2 points)** 8-bit 2's complement: __________________

3. Write the decimal value represented by the **bit pattern x88** in each of the following representations:

    a. **(2 points)** 8-bit unsigned: __________________

    b. **(2 points)** 8-bit 2's complement: __________________

Problem 1 (18 points), continued:

4. **(2 points)** Why can an N-bit unsigned representation represent larger integers than an N-bit 2's complement representation? Circle EXACTLY ONE ANSWER.

   a. The unsigned representation has more bits.

   b. It's a trick question. The maximum value represented is the same for both representations.

   c. Since some bit patterns are used for negative numbers with 2's complement, fewer bit patterns remain for positive numbers.

   d. Unsigned bits hold more information.

5. Prof. Lumetta needs your help. He wants to record the relationship between the ***N*** students taking ECE120 and the **four** lecture sections, but needs your advice on the number of bits needed. He is considering two different strategies, described below.

   a. **(2 points)** Strategy 1: For each lecture section, Prof. Lumetta uses 1 bit to represent whether each of the ECE120 students is in that section or not. Including all four lecture sections, how many bits does such a representation require?

      Answer: _______________

   b. **(2 points)** Strategy 2: Instead, Prof. Lumetta records which lecture section is attended by each of the ***N*** students, using the minimal number of bits needed per student. Including all ***N*** students, how many bits does such a representation require?

      Answer: _______________

Problem 2 (16 points): 2's complement

1. **(7 points)** What is the largest representable number in 8-bit 2's complement?

   a. Write that number in binary __________________________

   b. Give its decimal value __________________________

   c. What is the largest representable number in n-bit 2's complement? Express your answer as a function F(n).

   F(n) = _______________________

2. **(9 points)** Let **X = 1010 1100** and **Y = 1000 1001** be two 8-bit 2’s complement numbers. Show your work for the problems below**.**

   a. What is the **decimal value of X**? ________________
   (Express your answer as a simple number, not an expression: for example, write 23 rather than 21 + 2.)

   b. Compute **X - Y** using 2’s complement **addition**. Show your arithmetic, **including all carry bits (0's and 1's)**. (Leave your answer in 2’s complement binary form - not hexadecimal.) Does overflow occur?

   X - Y = __________________________

   Overflow? Yes ( ) No ( )

Problem 3 (18 points): Logical operations

1. **(4 points)** Perform the following bitwise logical operation. Express your answer in hexadecimal notation.

   x3B XOR xE9 = ____________________

2. **(8 points)** Answer the following related questions.

   a. Perform the following bitwise logical operation. Express your answer in binary.

   (1100 OR (NOT 1010)) = ____________________

   b. Now use your answer to part (a) to determine **all** 4-bit values of **Z** for which

   (1100 OR (NOT 1010)) AND NOT(Z) = 1001

   Write each value of **Z** in 4-bit binary form (for example, 1100): ________________

3. **(6 points)** Let C = $c_6c_5c_4c_3c_2c_1c_0$ be an alphabetic 7-bit ASCII character, chosen from either a-z (lower case) or A-Z (upper case). In this problem, you will show how to produce ASCII character D = $d_6d_5d_4d_3d_2d_1d_0$, the same letter with the opposite case.

   For example, for input C = 110 0111, which corresponds to lower case g, the output D must be 100 0111, which is the ASCII representation of upper case G. Similarly, for input C = 100 0111 (upper case G), the output D must be 110 0111 (lower case g).

   You can refer to the ASCII table provided on the last page of this exam.

   Specify the logical operation ♦ (AND, OR, NOT, NAND, NOR, or XOR) and the 7-bit constant $b_6b_5b_4b_3b_2b_1b_0$ so that $d_6d_5d_4d_3d_2d_1d_0 = c_6c_5c_4c_3c_2c_1c_0$ ♦ $b_6b_5b_4b_3b_2b_1b_0$.

   $c_6c_5c_4c_3c_2c_1c_0$ ____________________ ____________________

input	logical operation ♦	7-bit constant $b_6b_5b_4b_3b_2b_1b_0$

Problem 4 (15 points): Floating-point representation

In this problem, you can refer to the IEEE 754 single-precision format provided on the last page of this exam.

1. You will encode the decimal fraction $17\frac{5}{16}$ using IEEE 754 single-precision floating-point.

a. **(4 points)** Begin by converting $17\frac{5}{16}$ into binary.

Answer: ___________________________________

b. **(2 points)** Rewrite your answer to **part (a)** using normalized binary scientific notation, for example 1.001011×2^{13}.

Answer: **1.** __________________________ $\times$ **2**$^{\_\_\_\_\_\_}$

c. **(5 points)** Fill in the bits below to represent $17\frac{5}{16}$ in IEEE 754 single-precision floating-point. Some bits have been filled in for you already.

FIRST GROUP OF 16 BITS

								0	0	0	0	0	0	0	0

SECOND GROUP OF 16 BITS

Problem 4 (15 points), continued

2. (4 points) The number below is represented in IEEE 754 single-precision floating-point.

FIRST GROUP OF 16 BITS

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

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

SECOND GROUP OF 16 BITS

Divide the number by 16 and write the result in the boxes below, again using the IEEE 754 single-precision floating-point representation.

FIRST GROUP OF 16 BITS

SECOND GROUP OF 16 BITS

Problem 5 (16 points): C Program Analysis

A naïve programmer helped us with this problem, but forgot to write comments in the code to help you understand the purpose of the program. Alas, if only this naïve programmer would have paid attention to our advice!

Program

```c
#include <stdio.h>

int main ()
{
     char choice;
     int n, i;
     int result = 1;

     scanf("%c %d", &choice, &n);

     for (i=2; i<=n; i=i+1)
     {
          if ( choice=='s' )
          {
               result = result + i;
          }
          else if ( choice=='p' )
          {
               result = result * i;
          }
          else
          {
               printf("Happy Valentine's Day! ");
          }
     }
     printf("result=%d", result);

     return 0;
}
```

(The questions you need to answer can be found on the next page.)

Problem 5 (16 points), continued:

Given the C program on the previous page, write down the output EXACTLY as it will be printed on the screen when the program is executed and the user chooses to type the following values:

1. **(4 points)** The user typed `s 5`

   Answer: ____________________

2. **(4 points)** The user typed `p 4`

   Answer: ____________________

3. **(4 points)** The user typed `v 2`

   Answer: ____________________

4. **(4 points)** The user typed `p -2`

   Answer: ____________________

Problem 6 (17 points): C Program Completion

For this problem, we got help from another programmer who was careful to write comments for the following code. However, the programmer ran out of time writing the code. Thus, we need your help to complete the program below that computes an approximation to the number π using the following formula:

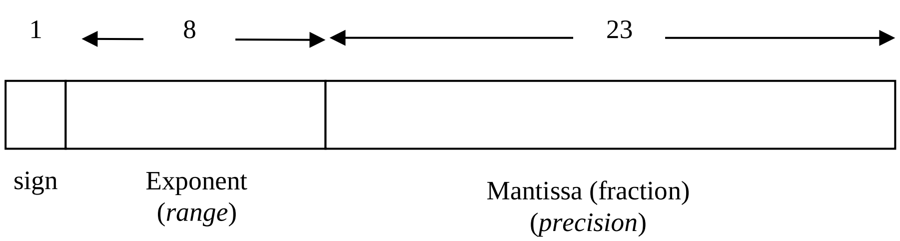
$$\pi \approx 2\prod_{i=1}^{n}\frac{4i^2}{(2i-1)(2i+1)} = 2\prod_{i=1}^{n}\frac{numerator(i)}{denominator(i)}$$

```
#include <stdio.h>

int main ()
{
     /* Define variables */
     int n, i;
     float numerator, denominator, pi;
     /* Initialize variable */
     float product = __________;
     /* User chooses value for variable n */
     ____________________ ("__________",__________);

     for (__________; __________; i=i+1)
     {
          /* Implement numerator(i) */
          numerator = ______________________________;
          /* Implement denominator(i) */
          denominator = ______________________________;
          /* Compute iteration */
          product = product * numerator/denominator;
     }

     /* Calculate value for pi */
     pi = ______________________________;
     /* Display result on screen */
     ____________________ ("PI=__________\n", __________);

     return 0;
}
```

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.