

**ECE 120 First Midterm Exam  
Spring 2017**

Tuesday, February 14, 2017

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

**KEY**

NetID: \_\_\_\_\_

Discussion Section and TA:

9:00 AM	<input type="checkbox"/>	AB1 Rui	
10:00 AM	<input type="checkbox"/>	AB2 Rui	
11:00 AM	<input type="checkbox"/>	AB3 Matt	
12:00 PM	<input type="checkbox"/>	AB4 Pawel	
1:00 PM	<input type="checkbox"/>	AB5 Pawel	
2:00 PM	<input type="checkbox"/>	AB6 Gowthami	<input type="checkbox"/> ABA Hui ren
3:00 PM	<input type="checkbox"/>	AB7 Gowthami	<input type="checkbox"/> ABB Hui ren
4:00 PM	<input type="checkbox"/>	AB8 Yu-Hsuan	<input type="checkbox"/> ABC Sifan
5:00 PM	<input type="checkbox"/>	AB9 Yu-Hsuan	<input type="checkbox"/> 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: 00101001

b. (2 points) 6-bit 2's complement to 8-bit 2's complement: 11101001

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

a. (2 points) 8-bit unsigned: 01011000

b. (2 points) 8-bit 2's complement: 01011000

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

a. (2 points) 8-bit unsigned: 136

b. (2 points) 8-bit 2's complement: -120

**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: 4N

- 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: 2N

**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 01111111

b. Give its decimal value 127

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

$$F(n) = \underline{2^{n-1}-1}$$

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**? -84  
(Express your answer as a simple number, not an expression: for example, write 23 rather than  $21 + 2$ .)

$$X = -(01010011+1) = -01010100 = -(2^6+2^4+2^2) = -36$$

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?

$$X - Y = \underline{00100011}$$

Overflow?    Yes (    )            No (X)

$$Y = -(01110110+1) = -01110111$$

$$X - (-Y) = X + Y = 10101100 + 01110111 = 00100011$$

**Problem 3 (18 points): Logical operations**

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

$$x3B \text{ XOR } xE9 = \underline{\text{xD2}}$$

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

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

$$(1100 \text{ OR } (\text{NOT } 1010)) = \underline{1101}$$

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

$$(1100 \text{ OR } (\text{NOT } 1010)) \text{ AND } \text{NOT}(Z) = 1001$$

Write each value of **Z** in 4-bit binary form (for example, 1100): 0110, 0111

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

$C_6C_5C_4C_3C_2C_1C_0$

XOR

0100000

input

logical operation  $\diamond$

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: 10001.0101

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

Answer: 1.00010101  $\times 2^{\underline{4}}$

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	1	1	1	1	0	1	1	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1	0	0	0	0	0	0	0	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

0	0	1	0	1	0	0	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

**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!

Line #	Program
1	#include <stdio.h>
2	
3	int main ()
4	{
5	char choice;
6	int n, i;
7	int result = 1;
8	
9	scanf("%c %d", &choice, &n);
10	
11	for (i=2; i<=n; i=i+1)
12	{
13	if ( choice=='s' )
14	{
15	result = result + i;
16	}
17	else if ( choice=='p' )
18	{
19	result = result * i;
20	}
21	else
22	{
23	printf("Happy Valentine's Day! ");
24	}
25	}
26	printf("result=%d", result);
27	
28	return 0;
29	}

(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: result=15

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

Answer: result=24

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

Answer: Happy Valentine's Day! result=1

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

Answer: result=1

### 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  $\pi$  using the following formula:

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

Line #	Program
1	#include <stdio.h>
2	
3	int main ()
4	{
5	/* Define variables */
6	int n, i;
7	float numerator, denominator, pi;
8	/* Initialize variable */
9	float product = <u>1</u> ;
10	/* User chooses value for variable n */
11	<u>scanf</u> (" <u>%d</u> ", <u>&amp;n</u> );
12	
13	for ( <u>i=1</u> ; <u>i&lt;=n</u> ; i=i+1 )
14	{
15	/* Implement numerator(i) */
16	numerator = <u>4*i*i</u> ;
17	/* Implement denominator(i) */
18	denominator = <u>(2 * i-1) * (2 * i+1)</u> ;
19	/* Compute iteration */
20	product = product * numerator/denominator;
21	}
22	
23	/* Calculate value for pi */
24	pi = <u>2 * product</u> ;
25	/* Display result on screen */
26	<u>printf</u> ("PI= <u>%f</u> \n", <u>pi</u> );
27	
28	return 0;
29	}

