CSARCH2 Mock Long Exam 2 Part 1

by: Clive Jarel Ang, Sean Benedict Bernardo, Brent Isaac Uy,
Enzo Arkin Panugayan, and Roan Cedric Campo

Important Reminders:

1. Read ALL instructions carefully and thoroughly before answering this mock exam.
2. The use of calculators and other computing devices are NOT allowed in the exam. However, you will be answering this in the comfort of your own home, so I literally have 0 control over the enforcement of that rule. ¯_(ツ)_/¯
3. Cheating in ANY form during the actual exam will be considered a major offense, merit you a 0.0 in the course, and would result in both Sir Rog and I becoming very sad.
4. This exam is GOOD FOR 4 HOURS. To be sufficiently prepared for the long exam proper, try to finish this mock exam in a shorter amount of time (while keeping a high score obv).
5. Yes, I'm sadistic and this mock exam reflects that. -Clive

I don't think Clive knows what Sadism means. -Enzo

i had fun making this :D -Sean

shout out fraser -Brent

I. Understanding Decimal Floating Point Representation

Express the following using IEEE 754 Single Precision (Decimal-32) format:

1. $26,884,291 \times 10^{72}$
2. $-94,722 \times 10^{392}$
3. $174,941 \times 10^{-70}$
4. $9,251,416 \times 10^{-43}$

#	Sign	Combi. Bit	Exp. Cont. Bit	Coefficient Continuation Bits	
1.	0	10010	101110	100 000 1 110	010 001 1 011
2.	1	11111	NaN	NaN	NaN
3.	0	00000	011111	001 111 0 100	001 100 1 101
4.	0	11001	111010	010 101 0 001	100 001 0 110

Express the following using IEEE 754 Double Precision (Decimal-64) format:

1. $132,981,718 \times 10^{141}$
2. $-1,122,334.455667788 \times 10^{-333}$
3. $891,112,289,184.7148 \times 10^{400}$

#	Sign	Combination Bits	Exponent Continuation Bits		
1.	0	10000	0001 1011		
	Coefficient Continuation Bits				
	000 000 0 000	000 000 0 000	001 011 0 010	001 100 1 111	111 001 1 000

#	Sign	Combination Bits	Exponent Continuation Bits		
2.	1	00001	0011 1000		
	Coefficient Continuation Bits				
	001 010 0 010	011 011 0 100	100 101 0 101	110 110 0 111	111 100 1 110

#	Sign	Combination Bits	Exponent Continuation Bits			
3.	0	11110	N/A			
	Coefficient Continuation Bits					
	N/A	N/A	N/A	N/A	N/A	

II. Understanding Unicode Representation #1

- Convert the Unicode to its equivalent UTF representation.
- The final answer should be in hexademical.
- Write "N/A" if not possible.

	U+DEC3
UTF-8 (answer format: xx xx xx xx or less is okay)	ED BB 83
UTF-16 (answer format: xxxx xxxx)	DEC3
UTF-32 (answer format: xxxx xxxx)	0000 DEC3

	U+9D83E
UTF-8 (answer format: xx xx xx xx or less is okay)	F2 9D A0 BE
UTF-16 (answer format: xxxx xxxx)	DA36 DC3E
UTF-32 (answer format: xxxx xxxx)	0009 D83E

III. Understanding Unicode Representation #2

- Write the Unicode equivalent of the given UTF representation.
- The final answer should be in hexademical.
- No need to write the "U+" prefix.
- Write "N/A" if not possible.

UTF Representation (encoding)	Unicode Code Point Equivalent
1) F7 8A AF 81 (UTF-8)	1CABC1
2) DA49 DC57 (UTF-16)	A2457
3) DBEA DEAA (UTF-16)	10AAAA
4) E0 A2 97 (UTF-8)	0897
5) 0010 DECB (UTF-32)	10DECB

IV. Understanding BCD Representation #1

- Determine the equivalent densely packed BCD representation.
- If there is no equivalent, write "N/A".

Decimal	DPBCD
413	100 001 0 011
927	1110101101
3,898	000 000 0 011 000 111 1 110
773	111 111 0 011
509,939	1010001001 0110111111
534	1010110100
8,928	000 000 1 000 011 010 1 110
67,953	0001100111 0111011101

V. Understanding BCD Representation #2

- Determine the equivalent decimal number.
- If there is no equivalent, write "N/A".

10-bit representation	2's Complement	Unsigned Integer	Densely Packed BCD
0011101110	238	238	988
1010101001	-343	681	529
1110011110	-98	926	996
0011000111	199	199	147
1111111111	-1	1023	999

VI. Understanding Memory Representation

- Determine the equivalent fixed-point decimal number or special case.
- Write "N/A" if it can't be represented.
- If applicable, specify special cases (+/- Infinity, sNaN, qNaN, Denormalized).

Internal Memory	View as ...	Fixed-point Decimal or Special Case Equivalent
0xFF800000	Binary32 floating point	-Infinity
0x40C1A9D000000000	Binary64 floating point	9043.625
0x7400000000000000	Decimal32 floating point	-1.0 * 10^-383
0x222C00000028F0CE	Decimal64 floating point	2934.188

VII. Designing a Data Type Representation

An IEEE-754/1985 quadruple-precision (128-bit) binary floating-point uses the following rules:

- 1-bit sign, 15-bit exponent representation, 112-bit significand
- Implicit binary normalization

1.) What is the bias or excess for E'?	16383
2.) What is the largest exponent (normalized)?	16383
3.) What is the smallest exponent (normalized)?	−16382
4.) What is the E' representation for infinity (in binary with ellipsis allowed)?	1..1
5.) What is the E' representation for quiet NaN (in binary with ellipsis allowed)?	1..1