Name: Michael Peters

Instructions:

- Download a version of this document
- Update the document with the required information
- Save the document
- Upload the document as your submission to this lab.

ASCII:

In class, we learned that the ASCII character set is a simple encoding. A hundred and twenty-seven characters were assigned a unique number. This number is known as its encoding. Correspondingly, when you are provided with an encoded value, you can decode the value to obtain the corresponding ASCII character.

For each row in the following table, complete the missing information by either encoding the ASCII character, decoding the numerical value, or by converting a numerical value into a different base.

ASCII	binary	octal	decimal	hexadecimal
a	01100001	141	97	61
D	01000100	104	68	44
LF	00001010	012	10	0A
EOT	00000100	004	4	04
(	00101000	050	40	28
K	0110 1011	153	107	0x6B
~	01111110	176	126	7E

You can use the following as resources:

1. the ASCII table that is provide via the man page on ssh.sandbox.csun.edu: man ascii
2. the gdb debugger on ssh.sandbox.csun.edu
    - $ ssh ssh.sandbox.csun.edu
    - $ gdb
    - (gdb) print /t 's'
    - (gdb) print /c 0x3D
    - (gdb) quit

UTF-8 Encoding

The professor provided an algorithm to convert a UTF-8 character, e.g., U+043F to its binary encoding. Use this algorithm to complete the following table.

UTF-8	Name / Character	binary	octal	decimal	hexadecimal
U+205D	⁝	1110 0010 1000 0001 1001 1101	70500635	14,844,317	0xe2819d
U+003D	Equal Sign	[illegible]	75	61	0x003D
U+03F0	ϰ	1100 1111 1011 0000	147660	53,168	0xcfb0
U+10D32	HANIFI ROHINGYA DIGIT TWO	1111 0000 1001 0000 1011 0100 1011 0010	360441322624	4,036,015,282	0xf090b4b2

BINARY → 0011 1101

Show your work each step of the way:

1. U+205D, e2819d
2. 0x205D
3. 10[illegible]0000 1011101
4. Byte 1: [illegible] 1110 xx10
   Byte 2: 10 000001
   Byte 3: 10 011101
5. 1110 0010 1000 0001 1001 1101
   13 0016 8401
6. 13

1. U+0030
   0030
2. 0x0030
3. 00000/0000 00[illegible]
   00[illegible] 110-1
4. Byte 1: 1110 0000
   Byte 2: 10 000000
   Byte 3: 10 000011
   10 11 1101

1. U+03F0
   U+03F0
2. 0xcfb0
3. 1100/1111/1011/0000
   Byte 1: 1110/1100
   Byte 2: 10 11/1110
   Byte 3: 10 11/0000
   1110/1100
   1011/1110/1011/0000

1. U+10D32
2. 0xf090b4b2
3. BINary
4. UTF

1110/0000/1000/0000/1011/0100

Base 10 Complements:

Provide the 10's and 9's complement for each of the following numbers with respect to 100,000.

Decimal	9's complement	10's complement*
54	99945	99946
45	99954	99955
145	99854	99855
255	99744	99745
34	99965	99966
195	99804	99805

- Note the 10's complement can be computed by adding 1 to the 9's complement

Binary Complements:

Provide the 2's and 1's complement for each of the following numbers represented with 8 bits.
Free free to use gdb to obtain the binary representation of each number

Decimal	Binary	1's complement	2's complement*
54	0011 0110	1100 1001	1100 1010
45	0010 1101	1101 0010	1101 0011
145	1001 0001	0110 1110	0110 1111
255	0000 1111 1111	1111 0000	1111 0000 0001
34	0010 0010	1101 1101	1101 1100
195	1100 0011	0011 1100	0011 1101

- Note the 2's complement can be computed by adding 1 to the 1's complement

Scientific Notation

Convert the following real numbers to Scientific Notation

Decimal	Scientific Notation		
	Mantissa	Base	Exponent
3.14	3.14	x 10	0
10.10	1.010	x 10	1
0.003765	3.765	x 10	-3
32000123.34	3.2000123 34	x 10	7

Scientific Notation

Convert the following real numbers to Scientific Notation. Feel free to use gdb to convert the exponent to binary (e.g., print /t 4 → 100)

Binary	Scientific Notation		
	Mantissa	Base	Exponent
10101.1011	1.01011011	x 2	100 (4)
111.11101	1.1111101	x 2	10 (2)
0.0010101	1.0101	x 2	-11 (-3)
10000101.011	1.00001011	x 2	111 (7)

Unary Numbering System:

[redacted]

Encode the following numbers using the Unary Numbering System. For all answers include the final stop bit.

Decimal	Unary Number
(4)₁₀	11110
5	111110
8	111111110
23	111111111111111111111110
435	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111…

111111111111111111111111111111111111111111111111111111111111…, etc. (435 and then 0)

Mathematical Review:

Add the following base 10 numbers together. Show all your work, and make sure you also provide the appropriate values of the status bits

- 365 + 36
- 3486 + 6666

					0	1	1		
						3	6	5	
				+		0	3	6	
						4	0	1	
Z	S	O	C						

					1	1	1	
					3	4	8	6
				+	6	6	6	6
			1		0	1	5	2
Z	S	O	C					

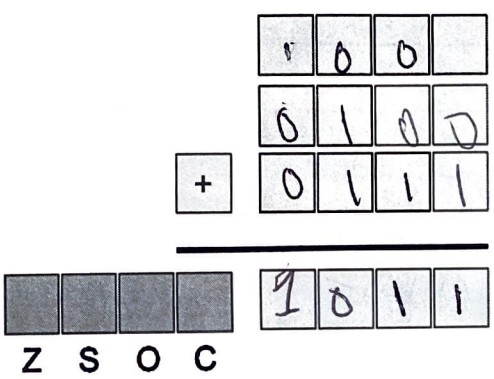
Binary Addition

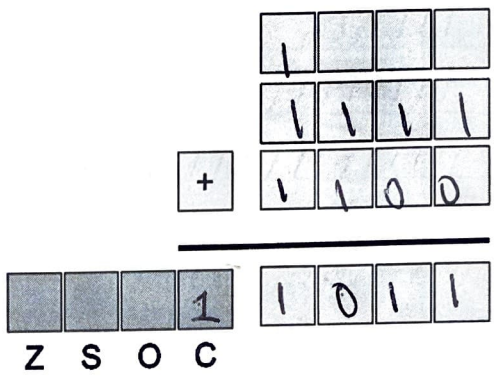
Add the following base 10 numbers together using binary addition. Show all your work, and make sure you also provide the appropriate values of the status bits. Your first step is to convert the base 10 numbers to binary. (Hint: be lazy, use gdb's print command.)

- 4 + 7
- 15 + 12

0100
0111

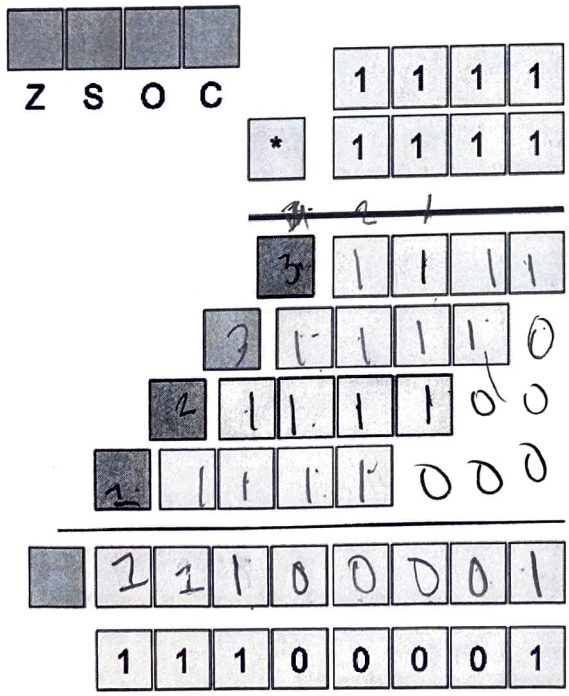
8421





Binary Multiplication

Perform the binary multiplication of 15 x 15.



15

15

225