

CS 261 Machine Organization Spring 2019**HW1: Computer Systems and Arithmetic (100 points)**

This assignment is due via GradeScope.

For part A you need to submit your `hwk1a.c` file that contains JUST your function to the `hw1a` submission on gradescope where 15/20 points will be immediately autograded. The remaining 5 points will be autograded as well, but not visible until after the due date.

For parts B and C you need to fill in the table in the fillable pdf form and submit the entire pdf form to the `hw1b` submission on gradescope. You can include your extra work after the tables. We have setup the submission to expect 6 pages after the table. If you didn't want to show your work that is fine, then include 6 blank pages.

A: Write a C function `hwk1a`, that takes a long int `x` as argument as well as two integers `n` and `m`, and returns a long int. Here is the function declaration:

```
long int hwk1a (long int x, int n, int m);
```

The function should swap byte `n` and `m` of a long int `x` (64-bit integer). A byte is an eight-bit aggregation. For this problem the index of the least significant byte is 0, and the index of the most significant byte 7 (so $0 \leq m, n \leq 7$). So, if `x = 0x0123456789ABCDEF` (a 64-bit integer), and if you swap two bytes say that are at the 3rd and 5th index, `x` would now be `x=0x0123896745ABCDEF`.

You are not allowed to use division, multiplication, or modulus, relative comparisons (`<`, `>`, `<=`, `>=`), loops, switches, function calls, macros, conditionals (if or `?:`)

You are allowed to use all bit level and logic operations, left and right shifts, addition and subtraction, equality and inequality tests, integer constants (`<=255`), `INT_MIN` and `INT_MAX`, and casting between data types. (20 points).

B: Fill in the missing values in the following table. The values should be in two's complement form and be 16 bits. (40 points, 2 points for each missing value)

Decimal	Binary	Hexadecimal
15	0b00000000000001111	0x000F
-15	0b1111111111110001	0xFF1
165		
	0b1110000011110001	
	0b0011101111011100	
		0xA5BC
-345		
		0x5CFB
	0b0010110110111100	
		0xD137
8024		
		0x1C5B

C: Fill the missing values in the following table. The values should be expressed in two's complement form and have 8 bits before the binary decimal point and 8 bits after the binary decimal point. For the decimal version you should write as a mixed fraction) (40 points, 2 points for each missing value)

Decimal	Binary	Hexadecimal
18 1/8	0b00010010.00100000	0x12.20
-18 1/8	0b11101101.11100000	0xED.E0
	0b11100010.01110000	
		0x3A.92
		0xF4.CB
	0b01001011.11010110	
120 7/32		
-92 12/128		
	0b00101101.11001010	
		0x52.0C
71 13/64		
	0b01101010.00111001	

Useful Notes:

1. For A please pay attention to what you are allowed to use and what you are not allowed to use.
2. For B and C please show intermediate steps on following pages if you wish to get partial credits in case your answer is wrong. You can scan handwritten work and merge back into the pdf you submit,

