Common NASM Mnemonics used in our Labs

Data Transfer			
move	**mov**	A, B	
load effective address	**lea**	A, B	
push reg onto stack	**push**	A	
pop reg off of stack	**pop**	A	
Arithmetic			
addition	**add**	A, B	
subtraction	**sub**	A, B	
multiply	**imul**	A	assumes RDX:RAX
divide	**idiv**	A	assumes RDX:RAX
propagate sign	**cqo**		affects RDX:RAX
Logic			
compare	**cmp**	A, B	
not	**not**	A	
and	**and**	A, B	
or	**or**	A, B	
exclusive or	**xor**	A, B	
Flow Control			
loop back while rcx>0	**loop**	address tag	
increment by 1	**inc**	A	
decrement by 1	**dec**	A	
unconditional branch	**jmp**	address tag	
jump if equal	**je**	address tag	
jump if not equal	**jne**	address tag	
jump if zero	**jz**	address tag	
jump if not zero	**jnz**	address tag	
jump if low	**jl**	address tag	
jump if not low	**jnl**	address tag	
jump if greater	**jg**	address tag	
jump if greater or equal	**jge**	address tag	
jump if overflow	**jo**	address tag	
jump if no overflow	**jno**	address tag	
Transfer Control			
call a procedure	**call**	procedure_name	
return from procedure	**ret**		
store to stack	**push**	register_name	updates RSP
load from stack	**pop**	register_name	updates RSP
Elasticity			
no operation	**nop**		