x86 MASM Assembly Cheat Sheet

[label:] mnemonic [operands] [; comment]

Mnemonic	Operands	Description
mov	dest, src	Move (copy) contents of src into dest
add	y, x	Add contents of x to y (result in y)
sub	y, x	Subtract x from y (result in y)
mul	x	Multiply eax by x (result in eax)
div	x	Divide eax by x, result in eax, remainder in edx
xor	y, x	Bitwise exclusive or (Tip, to zero a register: xor eax, eax)
jmp	label	Jump unconditionally to label
loop	label	If ecx > 0 jump to label and decrement ecx
cmp	x, y	Compare x and y and set the flags register accordingly
je	label	Jump to label if result of previous cmp operation was equal
j[ne|l|le|g|...]	label	Jump to label if ↑ not equal, <, <=, >, etc...
push	x	Push x onto the top of the stack, decrement esp
pop	x	Take top value off stack, put into x and increment esp

Example:

```
.386
.model flat, stdcall
ExitProcess PROTO, dwExitCode:DWORD

.data
        x DWORD 1                       ; a DWORD is a “double word”
        y DWORD 2                       ; a word is two bytes
        z DWORD 3                       ; thus a dword is 32 bits
.code
main PROC
        mov ebx, x
        mov ecx, y
        mov eax, z
        call sum3                       ; call the procedure
        invoke ExitProcess, eax         ; result is 1+2+3=6
main ENDP

sum3 PROC
        add eax, ebx
        add eax, ecx
        ret                             ; need to end procedures (other than main) in ret
sum3 ENDP

END main
```

C Programming Cheat Sheet

In case you haven't programmed in C before (but assuming you've programmed in something like python…

A function looks like this:

```
int sumTo(int n) {
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += i;
    }
    return sum;
}
```

Alternately, you might use a while loop. The following snippet is semantically identical to the above for-loop:

```
    {
        int i= 0;
        while(i < n) {
            sum += i;
            i++;
        }
    }
```

You'll need a main function, as in the Assembler we are simply going to return the result of our program from main:

```
int main() {
    return sumTo(100);
}
```