x86 Assembly Language: Live Coding Exercises

CMSC 313
Sections 01, 02

Challenge 1

```
eax = ebx + ecx
```


Challenge 2

```
// foo, fum and fee are in memory
// also allocate space for them
// in the data section

int foo = 0, fum = 0, fee = 0;

fee = foo + fum;
```


Challenge 3

```
// array "buf" is in memory
// allocate space for it in the
// .bss section

char buf[256];

edi = &buf[0];  // address of buf
*edi++ = 'a';
*edi = '-';
*++edi = 'z';
```

6

Challenge 4

```
if (eax < 0)
{
    eax++;
}
```

8

Challenge 5

```
if (eax > 0) {  /* eax is signed */
    eax++;
} else {
    eax = 0;
}
```

10

Challenge 6

```
ebx = 0;
while (eax > 0) {  /* eax is signed */
    ebx += eax;
    eax--;
}
```

12

Challenge 6b

```
ebx = 0;
do {
    ebx += eax;
    eax--;
} while (eax > 0); /* eax is signed */
```

14

A Bigger Example

- Now, let's look at toupper.asm in detail… here

16

Challenge 7

```
if (eax IS NEGATIVE AND ODD) {
    eax *= 2;
} else {
    eax /= 2;
}
```

17

Challenge 7b

```
if (eax IS NEGATIVE OR ODD) {
    eax *= 2;
} else {
    eax /= 2;
}
```

19

Challenge 8

```
/* eax is signed */
for (eax = 0; eax < ebx; eax++) {
    ebx -= eax;
    if (ebx < 0)
        break;
}
```

30

Challenge 9

You have a sequence of chars (bytes) at location "array", i.e.:

```
array: resb 64
```

which you wish to treat as a 8x8 array, in row-major order. You are given the requested row and column in EBX and ECX. Write a single "MOV" instruction to fetch that array element into AL

32