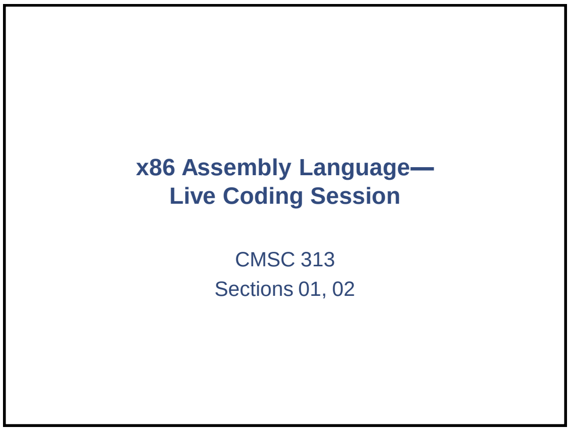
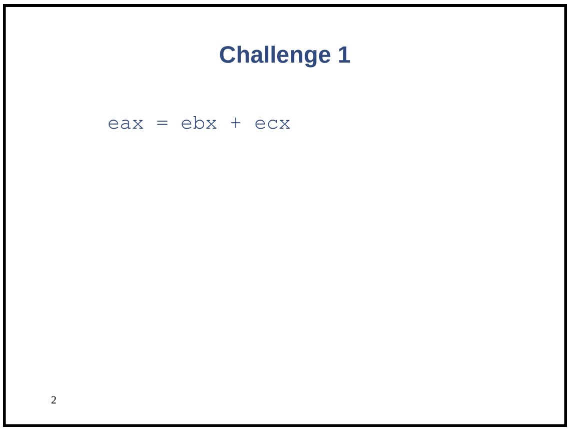


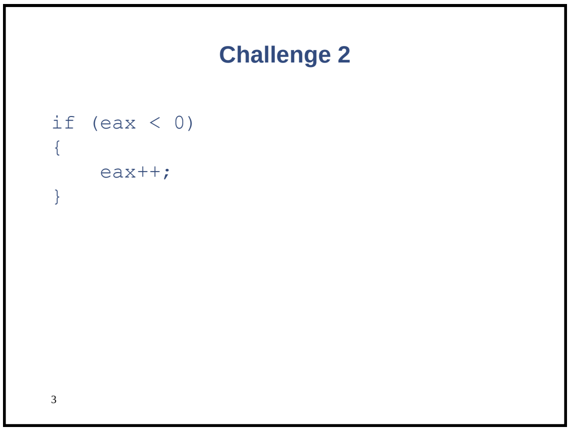
x86 Assembly Language—
Live Coding Session

CMSC 313
Sections 01, 02



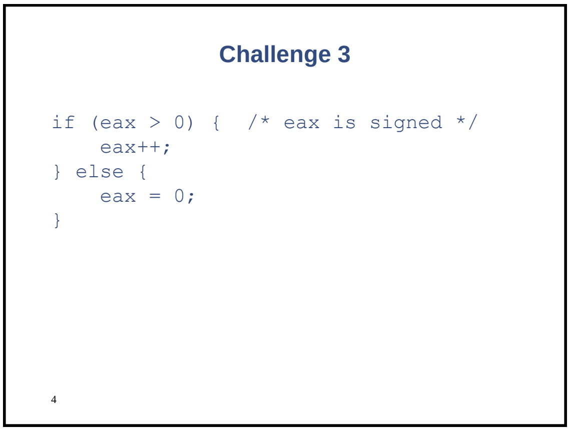
Challenge 1

```
eax = ebx + ecx
```



Challenge 2

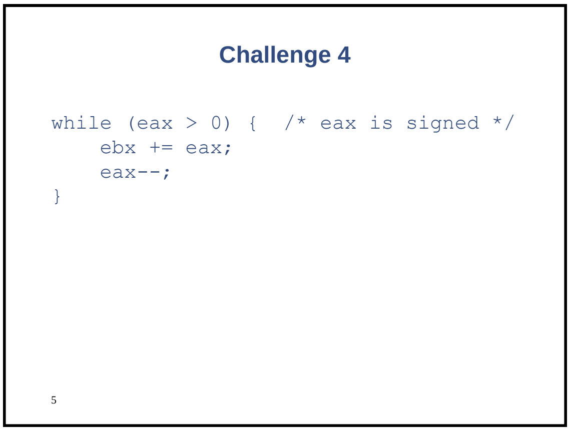
```
if (eax < 0)
{
    eax++;
}
```



Challenge 3

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

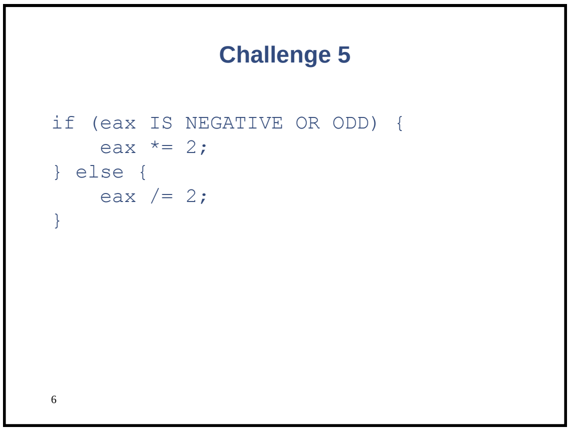
4



Challenge 4

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

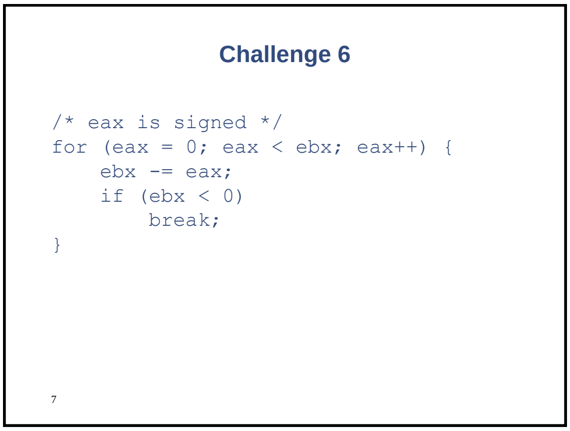
5



Challenge 5

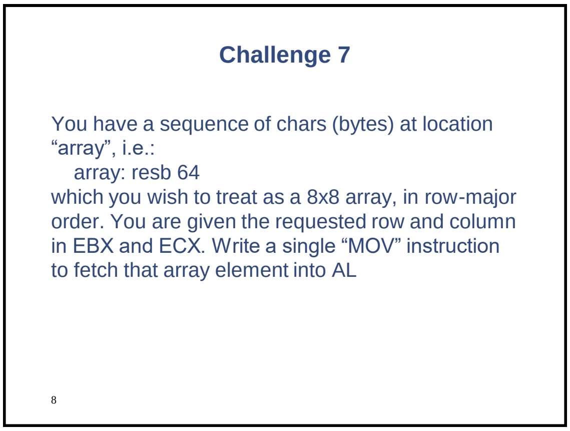
```
if (eax IS NEGATIVE OR ODD) {
    eax *= 2;
} else {
    eax /= 2;
}
```

6



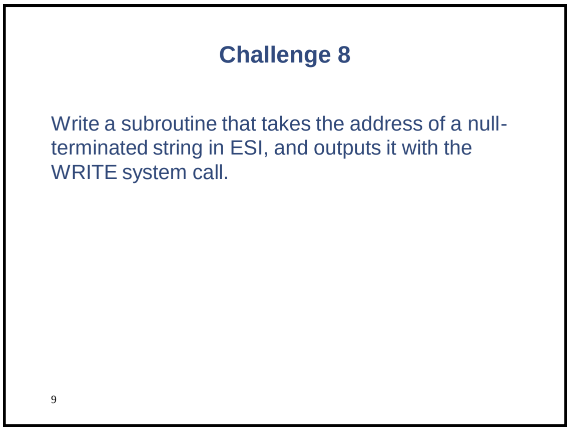
Challenge 6

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



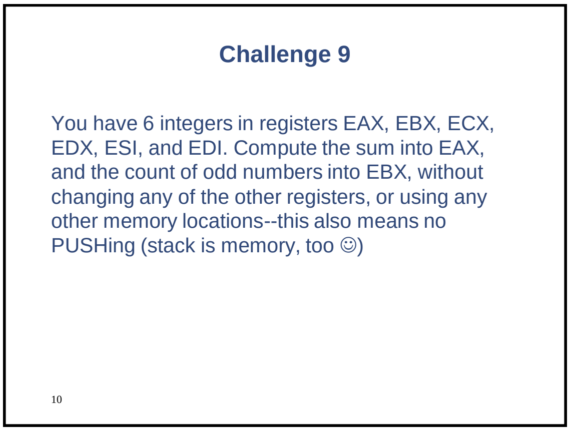
Challenge 7

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



Challenge 8

Write a subroutine that takes the address of a null-terminated string in ESI, and outputs it with the WRITE system call.



Challenge 9

You have 6 integers in registers EAX, EBX, ECX, EDX, ESI, and EDI. Compute the sum into EAX, and the count of odd numbers into EBX, without changing any of the other registers, or using any other memory locations--this also means no PUSHing (stack is memory, too ☺)

10