Check-in

Translate the following Jeff code to 3AC

```
fn : () int f{
  int a;
  a = 1 + 2;
  return a;
}
```

Administrivia

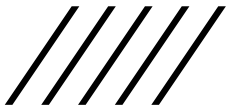
P5 due Friday

A couple of things you DON’T have to worry about:

- Checking open/close argument
- Checking the base or index of an array type

Oracle is live!

- It actually DOES handle the above cases if you want to follow them



F L I P P E D
W E D N E S D A Y

Written Work

Topics:

- 3AC
- ISAs

Question 1

Convert the following function into 3AC

```
int g;
int a(int b, int c){
        if (b){
                return 0;
        } else {
                b = b - 1 * c;
        }
        return b;
}
```

Question 2

Convert the following function into 3AC

```
int v(int a){
        while (a < 2){
                while (a < 3){
                        a++;
                }
                a++;
        }
        return a;
}
```

Question 3

Convert the following 3AC procedure into Jeff. Assume variables are local unless otherwise specified

```
enter k
        getarg 1, [b]
        [i] = [b]
lbl_1: [t] = [i] LT64 10
        ifz [v] goto lbl_2
        [i] = [i] ADD64 1
        WRITE [i]
        goto lbl_1
lbl_2: nop
leave k
```

Question 4

Assume a language that allows for pass-by-reference or pass-by-value parameters. What would the 3AC code look like for a pass-by-reference call? Illustrate with an example.

- Time Permitting: gdb for x64