1. Give pseudocode for a basic PID controller (without integrator anti-windup).
There are functions get_ref() and get_sensor() to call, and you can make others if you want.
There are already global variables, and you can add more:

```
static volatile float eint = 0;
static volatile float eprevious = 0;
```

The ISR is already setup to run at 1kHz:

```
__ISR(timer at 1kHz) {
        // your code here:
    err = get_ref() - read_sensor();
    dererr = err- eprevious;
    eprevious = err;
    eint = e – eint;
    u = kp*err+kd*dererr+ki*eint;
    send_response(u);

    interrupt_flag = 0;
}
```

2. Explain what integrator anti-windup is:

Integrator anti-windup is when the integral portion of the control is limited or not used when the error is too large so that the response does not oscillate which.

3. You have picked Kp, Ki, and Kd gains.
a. The response has too much overshoot. Which gain could you increase to reduce the overshoot?
Kd

b. The response has too much overshoot. Which gain could you decrease to reduce the overshoot?
Kp

c. The response has the right overshoot and settling time characteristics, but too much steady-state error. Which gain could you increase to reduce the steady-state error?
Ki