```cpp
#include "intake.h"

Intake::Intake(double port) : m(port)
{
  m.setBrakeMode(okapi::AbstractMotor::brakeMode::hold);
  m.getEncoder().reset();
}

void Intake::addPosition(int pos) {
  encPositions.push_back(pos);
}


//move four bar at full speed to position
void Intake::moveTarget(double enc) {
  m.moveAbsolute(enc, 200);
}


//set limits of four bar
void Intake::setLimits(int upper, int lower) {
  this->upper = upper;
  this->lower = lower;
  limits = true;
}


//run intake at speed while obeying limits
void Intake::run(bool left, bool right, double speed) {
  if(limits && ((m.getPosition()>upper && left) || (m.getPosition()<lower && right))) {
    m.moveVelocity(0);
  }
  else if(left) {
    m.moveVelocity(speed);
  }
  else if(right) {
    m.moveVelocity(-speed);
  }
  else if((!left && !right)) {
    m.moveVelocity(0);
  }
}

void Intake::handle(int count, double speed) {
  if(count%2 == 1 && count!= prevCount) {
    m.moveVelocity(-speed);
  }
  if(count%2 == 0 && count!= prevCount) {
    m.moveVelocity(0);
  }
}

void Intake::stepAbsolute(int count, double speed) {
  printf("count: %d\n", count % encPositions.size());
  if(prevCount != count) {
    double target = encPositions[count % encPositions.size()];
    m.moveAbsolute(target, speed);
```

```
  }
  prevCount = count;
}
```