

```
#include "main.h"
```

```
void autonZero(){  
    //Left  
    DRIVEBASE_POWER = 127;  
    setSyncMove(FORWARD, 400, false);  
    waitForTasks();  
    DRIVEBASE_POWER = 63;  
    setSyncMove(RIGHT, QUARTER, false);  
    waitForTasks();  
    openClaw();  
    waitForTasks();  
    DRIVEBASE_POWER = 127;  
    setSyncMove(FORWARD, 430, false);  
    waitForTasks();  
    closeClaw(300);  
    setSyncLift(HIGH_HEIGHT - 110);  
    waitForTasks();  
    DRIVEBASE_POWER = 63;  
    setSyncMove(LEFT, QUARTER, false);  
    waitForTasks();  
    DRIVEBASE_POWER = 63;  
    setSyncMove(FORWARD, 475, false);  
    waitForTasks();  
    openClaw();  
    setSyncLift(HIGH_HEIGHT + 40);  
}
```

```
void autonOne(){  
    autonZero();  
  
    waitForTasks();  
    DRIVEBASE_POWER = 127;  
    setSyncMove(BACKWARD, 300, false);  
    waitForTasks();  
    DRIVEBASE_POWER = 63;  
    setSyncMove(RIGHT, HALF + 60, false);  
    waitForTasks();  
    DRIVEBASE_POWER = 127;  
    setSyncMove(BACKWARD, 75, false);  
    setSyncLift(DOWN_HEIGHT);  
    waitForTasks();  
    setSyncMove(FORWARD, 350, false);  
    waitForTasks();  
    closeClaw(500);  
    setSyncMove(BACKWARD, 100, false);  
    waitForTasks();  
    setSyncLift(HIGH_HEIGHT);  
    waitForTasks();  
    DRIVEBASE_POWER = 63;  
    setSyncMove(LEFT, HALF, false);  
    waitForTasks();  
    DRIVEBASE_POWER = 127;  
    setSyncMove(FORWARD, 400, false);  
    waitForTasks();  
    setSyncMove(FORWARD, 75, false);  
    openClaw();  
    waitForTasks();  
}
```

```
void autonTwo(){  
    //Right square  
    setSyncMove(FORWARD, 400, false);  
    waitForTasks();  
    setSyncMove(LEFT, QUARTER, false);  
    waitForTasks();  
}
```

```

    openClaw();
    waitForTasks();
    setSyncMove(FORWARD, 475, false);
    waitForTasks();
    closeClaw(750);
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
    setSyncMove(RIGHT, QUARTER, false);
    waitForTasks();
    setSyncMove(FORWARD, 400, false);
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
    openClaw();
    setSyncLift(HIGH_HEIGHT - 20);
}

void autonThree(){
    //Right Square with Stars
    autonTwo();

    waitForTasks();
    setSyncMove(BACKWARD, 300, false);
    waitForTasks();
    setSyncMove(LEFT, HALF + 50, false);
    waitForTasks();
    setSyncMove(BACKWARD, 75, false);
    setSyncLift(DOWN_HEIGHT);
    waitForTasks();
    setSyncMove(FORWARD, 350, false);
    waitForTasks();
    closeClaw(750);
    setSyncMove(BACKWARD, 100, false);
    waitForTasks();
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
    setSyncMove(RIGHT, HALF + 100, false);
    waitForTasks();
    setSyncMove(FORWARD, 350, false);
    waitForTasks();
    openClaw();
    waitForTasks();
}

void autonFour(){
    //Left Anti-Middle
    setSyncMove(FORWARD, 100, false);
    waitForTasks();
    openClaw();
    waitForTasks();
    setSyncLift(HIGH_HEIGHT + 35);
    waitForTasks();
    setSyncMove(FORWARD, 750, false);
    waitForTasks();
    setSyncMove(RIGHT, THREE_QUARTER + 100, false);
    waitForTasks();
    setSyncLift(DOWN_HEIGHT);
    waitForTasks();
    setSyncMove(FORWARD, 750, false);
    waitForTasks();
    closeClaw(750);
    delay(750);
    waitForTasks();
    setSyncMove(BACKWARD, 250, false);
    waitForTasks();
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
}

```

```

        setSyncMove(LEFT, HALF + 100, false);
        waitForTasks();
        setSyncMove(FORWARD, 700, false);
        waitForTasks();
        openClaw();
        waitForTasks();
    }

    void autonFive(){
        //Left Anti-Middle
        setSyncMove(FORWARD, 100, false);
        waitForTasks();
        openClaw();
        waitForTasks();
        setSyncLift(HIGH_HEIGHT + 35);
        waitForTasks();
        setSyncMove(FORWARD, 750, false);
        waitForTasks();
    }

    void autonSix(){

    }

    void autonSeven(){

    }

    void autonEight(){

    }

    void autonNine(){

    }

    void autonTen(){

    }

    void autonEleven(){

    }

    void autonTwelve(){

    }

    void autonThirteen(){
        setSyncMove(FORWARD, 100, false);
        waitForTasks();
        openClaw();
        waitForTasks();
        taskDelay(200);
        setSyncLift(650);
        waitForTasks();
        setSyncMove(BACKWARD, 100, false);
        waitForTasks();
        taskDelay(1500);
        closeClaw(750);
        taskDelay(1500);
        setSyncMove(FORWARD, 950, false);
        waitForTasks();
        openClaw(); //Drop 3 star and cube combo
    }

```

```

waitForTasks();

setSyncMove(BACKWARD, 950, false);
waitForTasks();
taskDelay(1000);
closeClaw(750);
taskDelay(1000);
setSyncMove(FORWARD, 950, false);
waitForTasks();
openClaw();
waitForTasks(); //Drop the one cube preload

setSyncMove(BACKWARD, 475, false);
waitForTasks();
setSyncMove(RIGHT, QUARTER, false);
waitForTasks();
setSyncLift(25);
waitForTasks();
setSyncMove(FORWARD, 350, false);
waitForTasks();
closeClaw(400);
setSyncLift(625);
waitForTasks();
setSyncMove(FORWARD, 100, false);
waitForTasks();
taskDelay(400);
setSyncMove(LEFT, QUARTER, false);
waitForTasks();
setSyncMove(FORWARD, 475, false);
waitForTasks();
openClaw(); //Drop field cube

waitForTasks();
setSyncMove(BACKWARD, 275, false);
waitForTasks();
taskDelay(400);
setSyncMove(RIGHT, HALF + 50, false);
waitForTasks();
setSyncMove(BACKWARD, 100, false);
waitForTasks();
setSyncLift(25);
waitForTasks();
DRIVEBASE_POWER = 127;
setSyncMove(FORWARD, 350, false);
waitForTasks();
closeClaw(750);
setSyncMove(BACKWARD, 275, false);
waitForTasks();
setSyncLift(625);
waitForTasks();
DRIVEBASE_POWER = 63;
setSyncMove(LEFT, HALF + 50, false);
waitForTasks();
taskDelay(400);
setSyncMove(FORWARD, 350, false);
waitForTasks();
openClaw(); //Drop field stars

waitForTasks();
setSyncMove(BACKWARD, 400, false);
waitForTasks();
setSyncLift(50);
waitForTasks();
setSyncMove(FORWARD, 400, false);
waitForTasks();
closeClaw(750);

```

```

    setSyncMove(BACKWARD, 400, false);
    waitForTasks();
    setSyncLift(625);
    waitForTasks();
    setSyncMove(FORWARD, 500, false);
    waitForTasks();
    openClaw();
    waitForTasks(); //Drop fence stars
}

void autonFourteen(){
    setSyncMove(FORWARD, 100, false);
    waitForTasks();
    openClaw();
    waitForTasks();
    delay(200);
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
    setSyncMove(BACKWARD, 100, false);
    waitForTasks();
    delay(1500);
    closeClaw(750);
    delay(1500);
    setSyncMove(FORWARD, 950, false);
    waitForTasks();
    openClaw(); //Drop 3 star and cube combo
    waitForTasks();

    setSyncMove(BACKWARD, 950, false);
    waitForTasks();
    delay(1000);
    closeClaw(750);
    delay(1000);
    setSyncMove(FORWARD, 950, false);
    waitForTasks();
    openClaw();
    waitForTasks(); //Drop the one cube preload

    setSyncMove(BACKWARD, 1100, false);
    waitForTasks();
    gyroReset(gyroOne);
    gyroReset(gyroTwo);
    delay(1500);
    setSyncMove(FORWARD, 450, false);
    waitForTasks();
    delay(400);
    setSyncMove(RIGHT, -90, true);
    waitForTasks();
    setSyncLift(DOWN_HEIGHT);
    waitForTasks();
    setSyncMove(FORWARD, 500, false);
    waitForTasks();
    closeClaw(1000);
    delay(750);
    waitForTasks();
    setSyncLift(HIGH_HEIGHT);
    waitForTasks();
    setSyncMove(FORWARD, 500, false);
    waitForTasks();
    setSyncMove(LEFT, 0, true);
    waitForTasks();
    setSyncLift(HIGH_HEIGHT + 20);
    waitForTasks();
    setSyncMove(FORWARD, 500, false);
    waitForTasks();
    openClaw();

```

```

waitForTasks();

setSyncMove(BACKWARD, 500, false);
waitForTasks();
setSyncMove(RIGHT, -160, true);
waitForTasks();
setSyncMove(BACKWARD, 200, false);
waitForTasks();
setSyncLift(DOWN_HEIGHT);
waitForTasks();
setSyncMove(FORWARD, 600, false);
waitForTasks();
closeClaw(1000);
delay(1000);
setSyncMove(BACKWARD, 400, false);
waitForTasks();
setSyncLift(HIGH_HEIGHT + 30);
waitForTasks();
setSyncMove(LEFT, 0, true);
waitForTasks();
setSyncLift(DOWN_HEIGHT);
waitForTasks();
setSyncLift(HIGH_HEIGHT + 60);
waitForTasks();
setSyncMove(FORWARD, 1000, false);
waitForTasks();
openClaw();
waitForTasks();

setSyncMove(BACKWARD, 200, false);
waitForTasks();
setSyncMove(LEFT, 150, true);
waitForTasks();
setSyncLift(DOWN_HEIGHT);
waitForTasks();
setSyncMove(FORWARD, 350, false);
waitForTasks();
closeClaw(1000);
delay(1000);
setSyncMove(BACKWARD, 400, false);
waitForTasks();
setSyncLift(HIGH_HEIGHT + 20);
waitForTasks();
setSyncMove(RIGHT, 0, true);
waitForTasks();
setSyncMove(FORWARD, 700, false);
waitForTasks();
openClaw();
waitForTasks();
}

void autonomous() {
  autonSelection = programSelected(8);
  switch(autonSelection){
    case 0:
      autonZero();
      break;
    case 1:
      autonOne();
      break;
    case 2:
      autonTwo();
      break;
    case 3:
      autonThree();

```

```
break;  
case 4:  
    autonFour();  
break;  
case 5:  
    autonFive();  
break;  
case 6:  
    autonSix();  
break;  
case 7:  
    autonSeven();  
break;  
case 8:  
    autonEight();  
break;  
case 9:  
    autonNine();  
break;  
case 10:  
    autonTen();  
break;  
case 11:  
    autonEleven();  
break;  
case 12:  
    autonTwelve();  
break;  
case 13:  
    autonThirteen();  
break;  
case 14:  
    autonFourteen();  
break;  
default:  
    break;  
}  
  
}
```