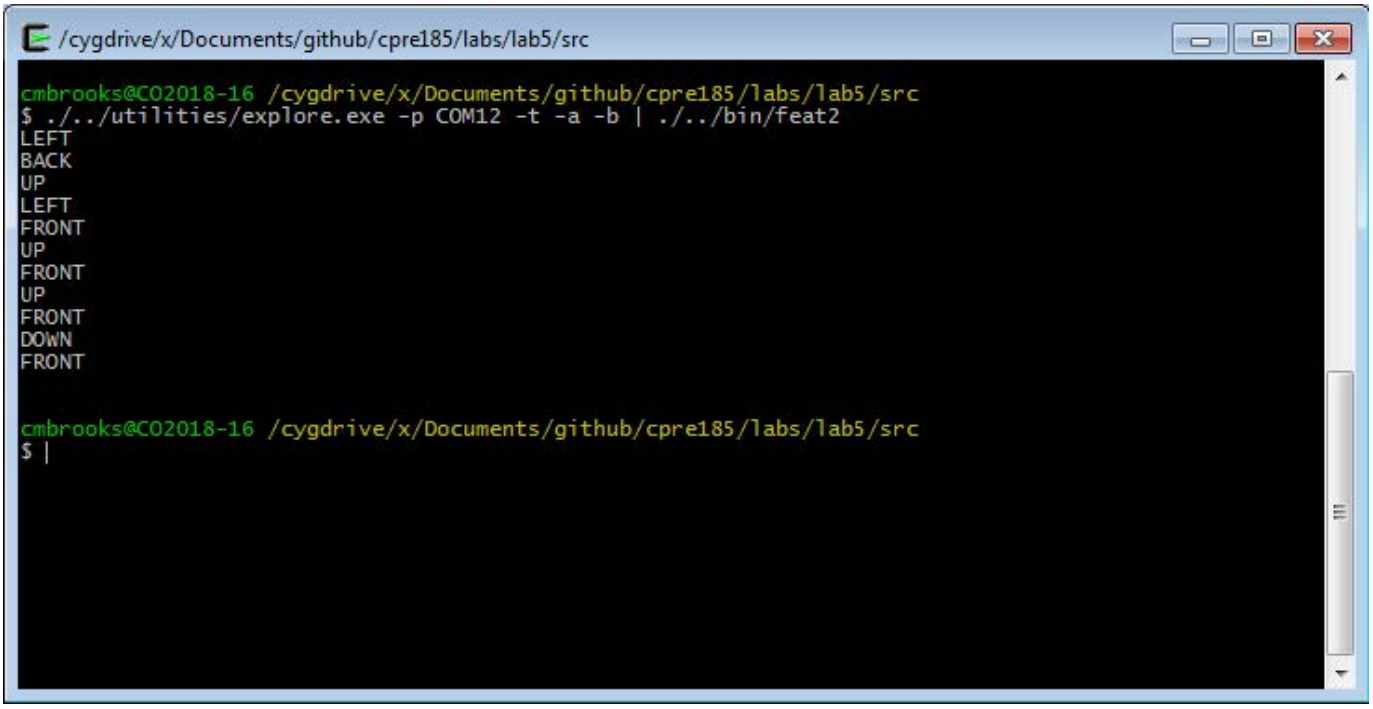
```c
/* Lab 5 Wrapper Program */

#include <stdio.h>
#include <math.h>
#include <stdbool.h>

/* Defines the possible orientations of the Esplora */
enum Direction {UP, DOWN, LEFT, RIGHT, FRONT, BACK};

/* Put your lab 4 functions prototypes here, as well as the prototype for lab 5 */
double mag (double x, double y, double z);
bool closeTo(double tolerance, double target, double val);
enum Direction getDirection(double x, double y, double z);


int main(void) {
    int t, b1, b2, b3, b4, b5, s;
    double ax, ay, az;
    enum Direction orientation;
    enum Direction lastOrientation;

    while (true) {
        scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3,
        &b4, &b5, &s );
        orientation = getDirection(ax, ay, az);

/* CODE SECTION 0
        printf("Echoing output: %d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d\n", t, ax, ay, az,
        b1, b2, b3, b4, b5, s);
*/

/*  CODE SECTION 1 */

        if (orientation != -1) {
            if (orientation != lastOrientation){

                if (b2 == 1){
                    return 0;  // Exits the program when the user presses the UP ARROW button
                    on the Esplora
                }

                if (orientation == UP) {
                    printf("UP\n");
                } else if (orientation == DOWN) {
                    printf("DOWN\n");
                } else if (orientation == LEFT) {
                    printf("LEFT\n");
                } else if (orientation == RIGHT){
                    printf("RIGHT\n");
                } else if (orientation == FRONT) {
                    printf("FRONT\n");
                } else if (orientation == BACK) {
                    printf("BACK\n");
```

```
                } else {
                    printf("MOVING\n");
                }

            }

            lastOrientation = orientation;

        }

    }

    return 0;

}

/* Put your lab 4 functions here, as well as your new function close_to */
double mag (double x, double y, double z) {
    double magnitude = sqrt(pow(x, 2) + pow(y, 2) + pow(z, 2));
    return magnitude;
}

bool closeTo(double tolerance, double target, double val) {
    double lower, higher;
    lower = (target - tolerance);
    higher = (target + tolerance);
    if (val > lower && val < higher) {
        return true;
    } else {
        return false;
    }
}

enum Direction getDirection (double x, double y, double z) {
    if (closeTo(0.25, 1, mag(x, y,z))) { // When the Esplora is not moving
        if (closeTo(0.25, 1, z)) {
            return UP;
        } else if (closeTo(0.25, -1, z)) {
            return DOWN;
        } else if (closeTo(0.25, 1, y)) {
            return FRONT;
        } else if (closeTo(0.25, -1, y)) {
            return BACK;
        } else if (closeTo(0.25, 1, x)) {
            return LEFT;
        } else if (closeTo(0.25, -1, x)) {
            return RIGHT;
        } else {
            return -1;
        }

    } else {
        return -1;
```

```
    }
}
```


/cygdrive/x/Documents/github/cpre185/labs/lab5/src
cmbrooks@CO2018-16 /cygdrive/x/Documents/github/cpre185/labs/lab5/src
$ ./../utilities/explore.exe -p COM12 -t -a -b | ./../bin/feat2
LEFT
BACK
UP
LEFT
FRONT
UP
FRONT
UP
FRONT
DOWN
FRONT
cmbrooks@CO2018-16 /cygdrive/x/Documents/github/cpre185/labs/lab5/src
$