

**Conditionals**

**LAB 5**

**SECTION G**

**SUBMITTED BY:**

**RYAN SAMUELSON**

**SUBMISSION DATE:**

**10/8/15**

## Problem

The problems was trying to create a piece of code that could tell what side was facing up for the X, Y, and Z coordinates. And have it split up in to functions and not have it not be jumble of code with everything mixed together. Another problem I faced in this lab was having it print out only one side and not repeat the side over and over again.

## Analysis

The problem states that only one side can be printed at a time and then if you switch sides and come back to the original side, have it print again. another problem I needed to focus on is have it print a side only when it stops moving, but that was easy to solve, because I just used the “man” function from the past labs to find the acceleration and then use that result in the “close\_to” function.

## Design

The way plan for stopping the repeating side names, was to use an “if” function to check if the value for the side has been printed, then after change the value to something so it can’t pass the “if” function. The second part to this problem was, after one side printed all the rest needed to be changed back to a value that could pass the “if” function. That is just what I did, I gave each side the ability to reset all of the other sides back to a value that could pass the “if” function, except for the side that just printed.

## Testing

In my testing I came across a problem that didn’t allow the side to print. I found that I made the code to strict and needed to simplify the parameters of the “if” functions to fit a wider impute range.

## Comments

In the screen shot I needed to test the code from my dorm, so I manually entered the “scan” values.

## Lab5

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

```
#define TRUE 1
#define FALSE 0
```

```
int close_to (double tolerance, double point, double value);
double mag(double ax, double ay, double az);
```

```
int main(void) {
    int t, b1, b2, b3, b4, b5, s;
    double ax, ay, az;
    double tolerance = .25;
```

```

double point = 1.0;

int hasTopPrinted = 0;
int hasBottomPrinted = 0;
int hasLeftPrinted = 0;
int hasRightPrinted = 0;
int hasFrontPrinted = 0;
int hasBackPrinted = 0;

while (TRUE) {
    scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3,
    &b4, &b5, &s );
    if(b2 == 1){
        break;
    }
    double x = mag(ax, ay, az);
    if (close_to (tolerance, point, x) == TRUE){
        if(az > .75){
            if (hasTopPrinted == 0) {
                printf("top\n");
                hasTopPrinted = 1;
                hasBottomPrinted = 0;
                hasLeftPrinted = 0;
                hasRightPrinted = 0;
                hasFrontPrinted = 0;
                hasBackPrinted = 0;
            }
        }
        else if(az < -.75){
            if (hasBottomPrinted == 0){
                printf("bottom\n");
                hasTopPrinted = 0;
                hasBottomPrinted = 1;
                hasLeftPrinted = 0;
                hasRightPrinted = 0;
                hasFrontPrinted = 0;
                hasBackPrinted = 0;
            }
        }
        else if(ax < -.75){
            if (hasLeftPrinted == 0){
                printf("left\n");
                hasTopPrinted = 0;
                hasBottomPrinted = 0;
                hasLeftPrinted = 1;
                hasRightPrinted = 0;
                hasFrontPrinted = 0;
                hasBackPrinted = 0;
            }
        }
    }
}

```

```

        }
    }
    else if(ax > .75){
        if (hasRightPrinted == 0){
            printf("right\n");
            hasTopPrinted = 0;
            hasBottomPrinted = 0;
            hasLeftPrinted = 0;
            hasRightPrinted = 1;
            hasFrontPrinted = 0;
            hasBackPrinted = 0;
        }
    }
    else if(ay > .75){
        if (hasFrontPrinted == 0){
            printf("front\n");
            hasTopPrinted = 0;
            hasBottomPrinted = 0;
            hasLeftPrinted = 0;
            hasRightPrinted = 0;
            hasFrontPrinted = 1;
            hasBackPrinted = 0;
        }
    }
    else if(ay < -.75){
        if (hasBackPrinted == 0){
            printf("back\n");
            hasTopPrinted = 0;
            hasBottomPrinted = 0;
            hasLeftPrinted = 0;
            hasRightPrinted = 0;
            hasFrontPrinted = 0;
            hasBackPrinted = 1;
        }
    }
}

}

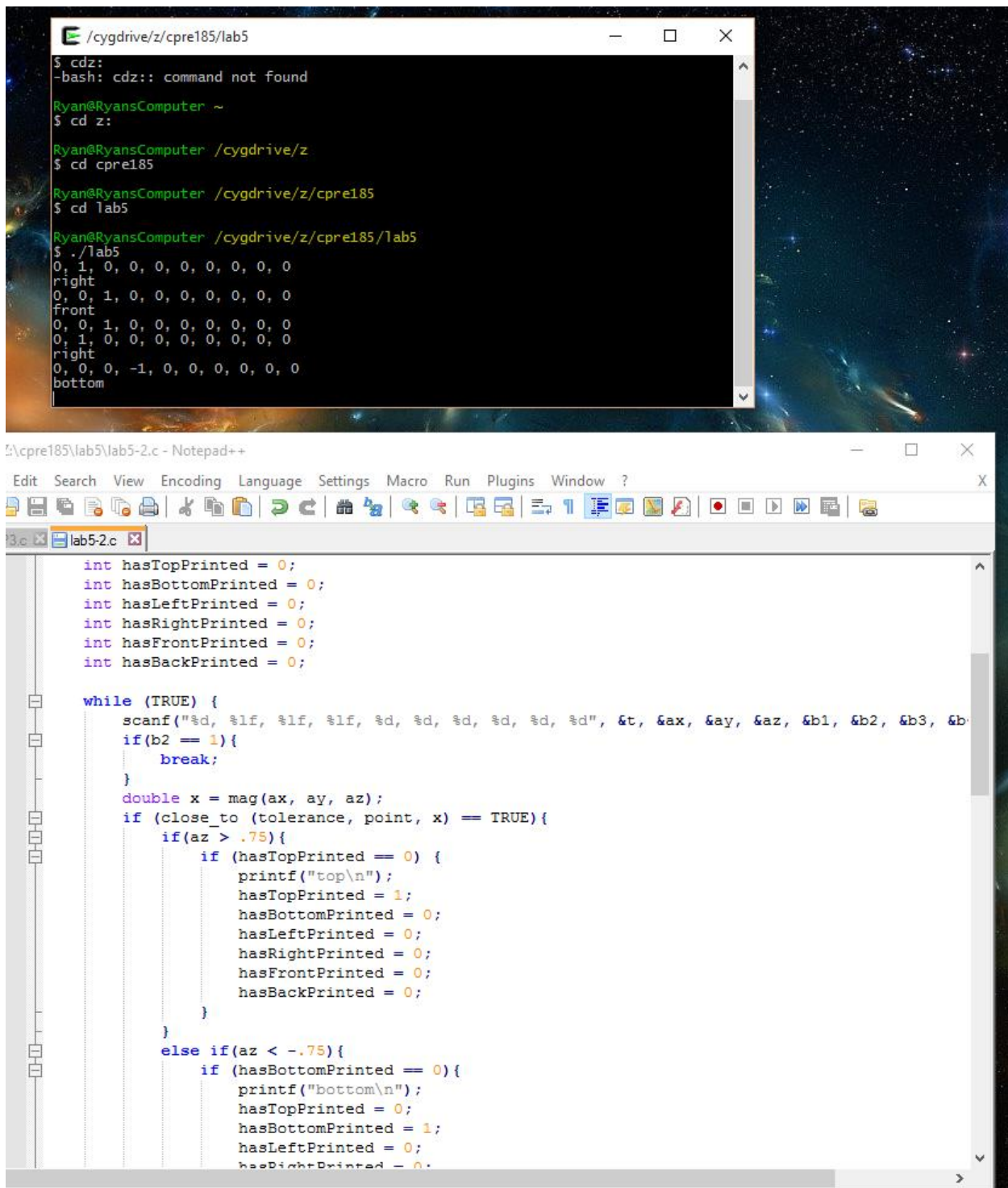
return 0;
}

double mag(double ax, double ay, double az){
    double r = sqrt(ax*ax+ay*ay+az*az);
    return r;
}

int close_to (double tolerance, double point, double value){
    if(point > value){
        if((point - value) < tolerance){
            return TRUE;
        }
    }
}

```

```
        }
        else {
            return FALSE;
        }
    }
    else {
        if((value - point) < tolerance){
            return TRUE;
        }
        else {
            return FALSE;
        }
    }
}
```



The image shows a Windows desktop environment with a space-themed wallpaper. Two windows are open:

- Terminal Window:** The title bar shows the path `/cygdrive/z/cpre185/lab5`. The command history is as follows:

```
$ cdz:
-bash: cdz:: command not found

Ryan@RyansComputer ~
$ cd z:

Ryan@RyansComputer /cygdrive/z
$ cd cpre185

Ryan@RyansComputer /cygdrive/z/cpre185
$ cd lab5

Ryan@RyansComputer /cygdrive/z/cpre185/lab5
$ ./lab5
0, 1, 0, 0, 0, 0, 0, 0, 0, 0
right
0, 0, 1, 0, 0, 0, 0, 0, 0, 0
front
0, 0, 1, 0, 0, 0, 0, 0, 0, 0
0, 1, 0, 0, 0, 0, 0, 0, 0, 0
right
0, 0, 0, -1, 0, 0, 0, 0, 0, 0
bottom
```
- Notepad++ Editor:** The title bar shows the path `z:\cpre185\lab5\lab5-2.c - Notepad++`. The code in the editor is as follows:

```
int hasTopPrinted = 0;
int hasBottomPrinted = 0;
int hasLeftPrinted = 0;
int hasRightPrinted = 0;
int hasFrontPrinted = 0;
int hasBackPrinted = 0;

while (TRUE) {
    scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3, &b4);
    if(b2 == 1){
        break;
    }
    double x = mag(ax, ay, az);
    if (close_to (tolerance, point, x) == TRUE){
        if(az > .75){
            if (hasTopPrinted == 0) {
                printf("top\n");
                hasTopPrinted = 1;
                hasBottomPrinted = 0;
                hasLeftPrinted = 0;
                hasRightPrinted = 0;
                hasFrontPrinted = 0;
                hasBackPrinted = 0;
            }
        }
        else if(az < -.75){
            if (hasBottomPrinted == 0){
                printf("bottom\n");
                hasTopPrinted = 0;
                hasBottomPrinted = 1;
                hasLeftPrinted = 0;
                hasRightPrinted = 0;
                hasFrontPrinted = 0;
                hasBackPrinted = 0;
            }
        }
    }
}
```