DS4 Equalizer

LAB 7

SECTION C

Akash Patel

SUBMISSION DATE: 3/28/19

3/28/19

Problem

In this lab the objective was to print a graph of lefts and rights wave depending on the pitch and roll of the controller. The pitch and roll should switch depending on which button is pressed. If the controller or joystick was in the center then it should print 0s.

Analysis

This program required a lot of case statements and formulas that would show where the controller was at. They kept track of characters and values that were in a certain range. The hardest part of the program was tracking the pitch and roll of the controller and the joystick.

Design

This lab was skeleton coded and basically set up for us. I just copied and pasted the prototypes and turned them into functions. I had to get values and write equations to get the program to compile and work properly. I first created the functions in the main however I had a lot of difficulty and problems with that so I went with what I have now. I made a true or false statement with the square button to stop the program. I also created pitch and roll functions to set parameters for the pitch and roll. I used -1 and 1 as the parameters, then returned the sin of them. I also had to print 40 characters and blank spaces depending on the pitch and roll so I made a function for that where it would take the number and see whether its less then or greater than one. Then I created the graph line function so it would print chracters L or R depending on the value.

Testing

At first I had a bunch of debug errors and I was also using the wrong flags so it wouldn't even run properly. Once I took out the -a it started working.

Comments

I was unable to correctly modify for it to print the joystick outputs. I tried many different methods but it just wouldn't work.

Questions:

1.) I scaled my values by setting the creating an equation that multiplied the value by 39 and the rad divided by (pi/2). This then gives a value that is a max 39 and min -39, however it gives a decimal value so I have to use int to truncate it to a int value. I did the same for joystick however I also multiplied it by 128 because the threshold is 128.

```
double xd = 39.0 * (rad / (PI/2));
        return (int)xd;
rad = -rad;
        double xd = 39.0 * ((double)rad / (double)128);
        return (int)xd;
```

2.) The graph behavior near its limits starts oscillating according to my program. It also shows the letters all the way to the end of each graph.

Source Code

<Use NPP Exporter to PASTE source code>

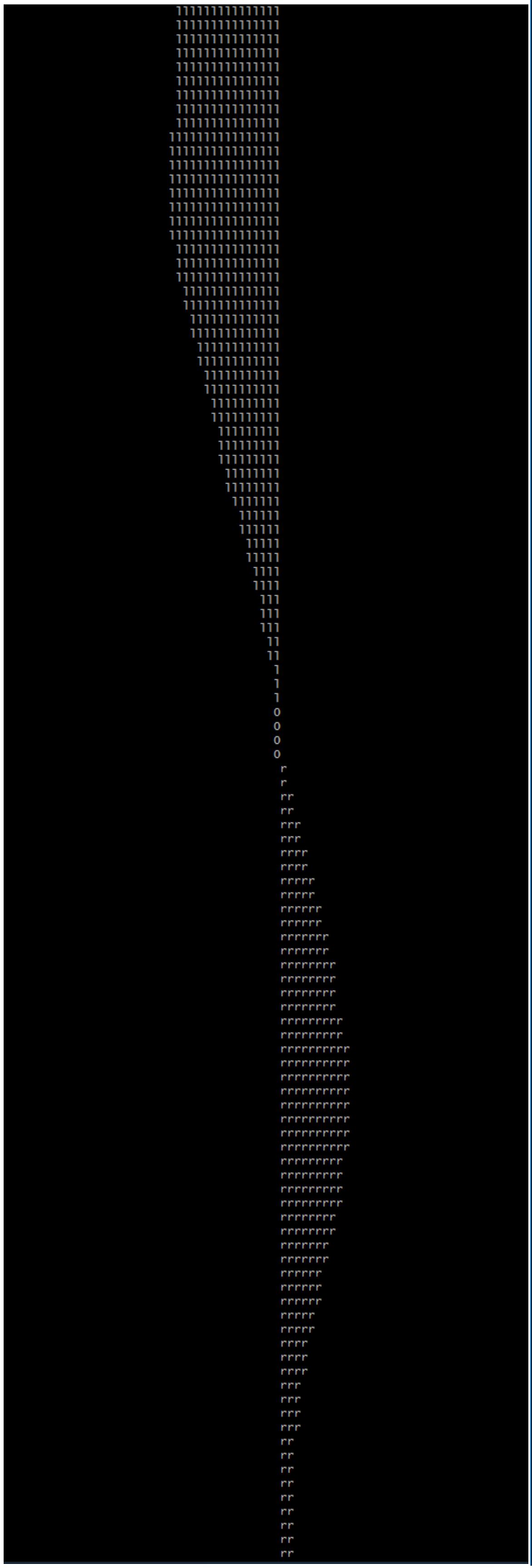
```
/*-----------------------------------------------------------------------------
-                                         SE 185 Lab 07
-             Developed for 185-Rursch by T.Tran and K.Wang
-      Name:
-      Section:
-      NetID:
-      Date:
-
-   This file provides the outline for your program
-   Please implement the functions given by the prototypes below and
-   complete the main function to make the program complete.
-   You must implement the functions which are prototyped below exactly
-   as they are requested.
-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------
-                                                   Includes
-----------------------------------------------------------------------------*/
#include <stdio.h>
#include <math.h>


/*-----------------------------------------------------------------------------
-                                                   Defines
-----------------------------------------------------------------------------*/
#define PI 3.141592653589

/* NO GLOBAL VARIABLES ALLOWED */


/*-----------------------------------------------------------------------------
-                                                   Prototypes
-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------------
-
    PRE: Arguments must point to double variables or int variables as appropriate
    This function scans a line of DS4 data, and returns
    True when left button is pressed
    False Otherwise
    POST: it modifies its arguments to return values read from the input line.
----------------------------------------------------------------------------------
*/
int read_input( int* time,
                double* g_x, double* g_y, double* g_z,
                int* button_T, int* button_C, int* button_X, int* button_S,
                int* l_joy_x, int* l_joy_y, int* r_joy_x, int* r_joy_y );

/*-----------------------------------------------------------------------------
    PRE: ~(-1.0) <= mag <= ~(1.0)
    This function scales the roll/pitch value to fit on the screen.
```

```c
    Input should be capped at either -1.0 or 1.0 before the rest of your
    conversion.
    POST: -39 <= return value <= 39
-----------------------------------------------------------------------------*/
int scaleMagForScreen(double rad);

/*-----------------------------------------------------------------------------
    PRE: -128 <= mag <= 127
    This function scales the joystick value to fit on the screen.
    POST: -39 <= return value <= 39
-----------------------------------------------------------------------------*/
int scaleJoyForScreen(int rad);

/*----------------------------------------------------------------------------
    PRE: -39 <= number <= 39
    Uses print_chars to graph a number from -39 to 39 on the screen.
    You may assume that the screen is 80 characters wide.
----------------------------------------------------------------------------*/
void graph_line(int number);

/*-----------------------------------------------------------------------------
    PRE: num >= 0
    This function prints the character "use" to the screen "num" times
    This function is the ONLY place printf is allowed to be used
    POST: nothing is returned, but "use" has been printed "num" times
-----------------------------------------------------------------------------*/
void print_chars(int num, char use);


/*-----------------------------------------------------------------------------
-                                                   Implementation
-----------------------------------------------------------------------------*/
int main()
{
    double x, y, z;                         /* Values of x, y, and z axis*/
    int t;                                  /* Variable to hold the time value */
    int b_Up, b_Down, b_Left, b_Right;      /* Variables to hold the button statuses */
    int j_LX, j_LY, j_RX, j_RY;             /* Variables to hold the joystick statuses */
    int scaled_pitch, scaled_roll;             /* Value of the roll/pitch adjusted to
fit screen display */
    int scaled_joy;                         /* Value of joystick adjusted to fit screen
display */

      int current = 0;
      int d = 0;
      int k = 0;



    /* Put pre-loop preparation code here */
   int read_input( int* time,
                double* g_x, double* g_y, double* g_z,
                int* button_T, int* button_C, int* button_X, int* button_S,
                int* l_joy_x, int* l_joy_y, int* r_joy_x, int* r_joy_y ){

```

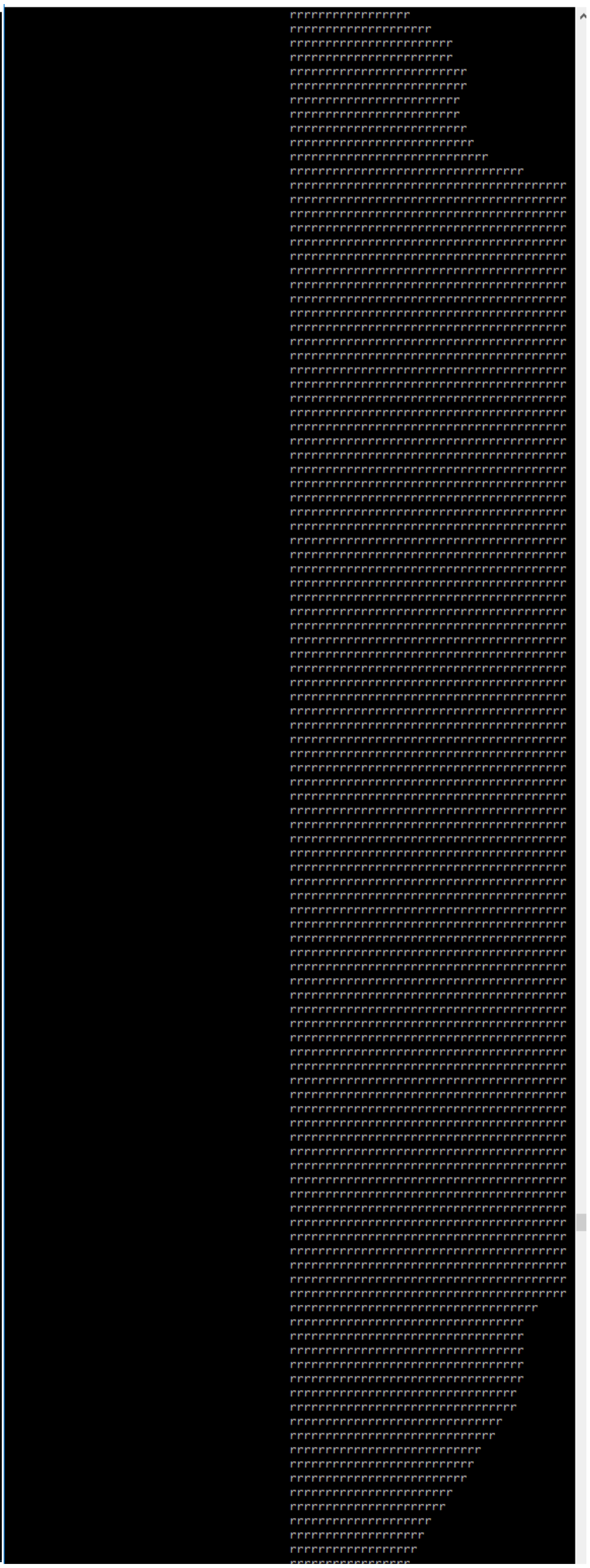
```
					scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d, %d, %d", time,
g_x, g_y, g_z, button_T, button_C, button_X, button_S, l_joy_x, l_joy_y, r_joy_x,
r_joy_y);
					if(*button_S == 1){
						return 1;
					}
					else{
						return 0;
					}
						}

/* Calculate and scale for pitch AND roll AND joystick */
int scaleMagForScreen(double rad){
	double xd = 39.0 * (rad / (PI/2));
	return (int)xd;
}


int scaleJoyForScreen(int rad){
	rad = -rad;
	double xd = 39.0 * ((double)rad / (double)128);
	return (int)xd;
}

 /* Switch between roll, pitch, and joystick with the up, down, and right button,
respectivly */
double roll(double g_x)
{
	if(g_x > 1.0)
	{
		g_x = 1.0;
	}
	else if(g_x < -1.0)
	{
		g_x = -1.0;
	}
		return asin(g_x);
}

double pitch(double g_y)
{
	if(g_y > 1.0)
	{
		g_y = 1.0;
	}
	else if(g_y < -1.0)
	{
		g_y = -1.0;
	}
		return asin(g_y);
}


void print_chars(int num, char use){
        int i = 0;
        int numChars;

        if(num < 0)
        {
                numChars = 40 + num;
                for (i = 0; i < numChars; i++)
                {
                        printf(" ");
                }

                for(i = num; i < 0; i++)
                {
                        printf("l");
                }
        }

        else if(num > 0)
        {
                for(i = 0; i < 40; i++)
                {
                        printf(" ");
                }
                for(i = num; i > 0; i--)
                {
                        printf("%c", use);
                }

                for(i = 0; i < (40 - num); i++)
                {
                        printf(" ");
                }
        }

        else if(num == 0)
        {
                for(i = 0; i < 39; i++)
                {
                        printf(" ");
                }
                printf("0");
        }
        printf("\n");
}


 /* Output your graph line */
void graph_line(int number, int num){
        char r,l;
        if(number < 0){
                print_chars(number, 'l');

        }
        else if(number > 0){
                print_chars(number, 'r');
        }
        else if(number == 0){
                print_chars(0, '0');
        }
}



    do
     {
                 /* Scan a line of input */
                current = read_input(&t, &x, &y, &z, &b_Up, &b_Right, &b_Down, &b_Left,
&j_LX, &j_LY, &j_RX, &j_RY );

                        if(b_Up == 1 && d > 200){
                                d = 0;
                                if(k==2){
                                        k=0;
                                }
                                else{
                                        k += 1;
                                }
                        }
                        d ++;
                                if (k == 0){
                                graph_line(scaleMagForScreen(roll(x)),1);
                                }
                                if (k == 1){
                                graph_line(scaleMagForScreen(pitch(y)),0);
                                }
                                if (k == 2){
                                graph_line(scaleJoyForScreen(b_Left),1);
                                }
          fflush(stdout);

     }
        while (current == 0); /* Modify to stop when left button is pressed */

     return 0;

        }
```

Screen Shots

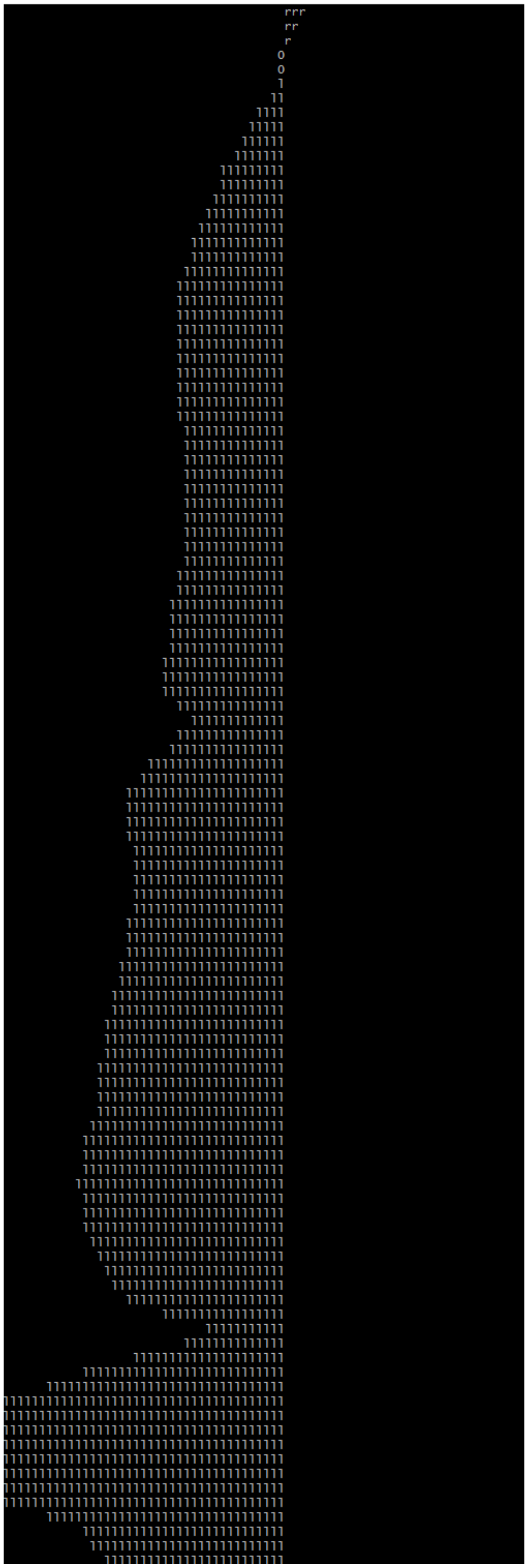
<Number the screenshots and paste here. The point of numbering the screenshots is so that you can refer to them during your discussion in the various parts above. Alternatively, you can include the screenshots in-line with the text above as part of your discussion.>



```
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
llllllllllllllll
 lllllllllllllll
 lllllllllllllll
 lllllllllllllll
  llllllllllllll
  llllllllllllll
   lllllllllllll
   lllllllllllll
    llllllllllll
    llllllllllll
     lllllllllll
     lllllllllll
      llllllllll
      llllllllll
       lllllllll
       lllllllll
       lllllllll
        llllllll
        llllllll
         lllllll
          llllll
          llllll
           lllll
           lllll
            llll
            llll
             lll
             lll
             lll
              ll
              ll
               l
               l
               l
               0
               0
               0
               0
                r
                r
                rr
                rr
                rrr
                rrr
                rrrr
                rrrr
                rrrrr
                rrrrr
                rrrrrr
                rrrrrr
                rrrrrrr
                rrrrrrr
                rrrrrrrr
                rrrrrrrr
                rrrrrrrr
                rrrrrrrr
                rrrrrrrrr
                rrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrrr
                rrrrrrrrr
                rrrrrrrrr
                rrrrrrrrr
                rrrrrrrrr
                rrrrrrrr
                rrrrrrrr
                rrrrrrr
                rrrrrrr
                rrrrrr
                rrrrrr
                rrrrrr
                rrrrr
                rrrrr
                rrrr
                rrrr
                rrrr
                rrr
                rrr
                rrr
                rrr
                rr
                rr
                rr
                rr
                rr
                rr
                rr
                rr
                rr
```



```
rrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrr
```



```
                                       rrr
                                       rr
                                       r
                                      0
                                      0
                                      1
                                     11
                                   1111
                                  11111
                                 111111
                                1111111
                              111111111
                              111111111
                             1111111111
                            11111111111
                           111111111111
                          1111111111111
                          1111111111111
                         11111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                        111111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                         11111111111111
                        111111111111111
                        111111111111111
                       1111111111111111
                       1111111111111111
                       1111111111111111
                       1111111111111111
                      11111111111111111
                      11111111111111111
                      11111111111111111
                        111111111111111
                          1111111111111
                        111111111111111
                       1111111111111111
                    1111111111111111111
                   11111111111111111111
                 1111111111111111111111
                 1111111111111111111111
                 1111111111111111111111
                 1111111111111111111111
                  111111111111111111111
                  111111111111111111111
                  111111111111111111111
                  111111111111111111111
                  111111111111111111111
                 1111111111111111111111
                 1111111111111111111111
                 1111111111111111111111
                11111111111111111111111
                11111111111111111111111
               111111111111111111111111
               111111111111111111111111
              1111111111111111111111111
              1111111111111111111111111
              1111111111111111111111111
             11111111111111111111111111
             11111111111111111111111111
             11111111111111111111111111
             11111111111111111111111111
            111111111111111111111111111
           1111111111111111111111111111
           1111111111111111111111111111
           1111111111111111111111111111
          11111111111111111111111111111
           1111111111111111111111111111
           1111111111111111111111111111
           1111111111111111111111111111
            111111111111111111111111111
             11111111111111111111111111
              1111111111111111111111111
               111111111111111111111111
                 1111111111111111111111
                      11111111111111111
                            11111111111
                         11111111111111
                  111111111111111111111
           1111111111111111111111111111
      111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
111111111111111111111111111111111111111
      111111111111111111111111111111111
           1111111111111111111111111111
            111111111111111111111111111
              1111111111111111111111111
```