Assignment 0
Left, Right and Center

Prof. Darrell Long
Department of Computer Engineering

Spring 2017

Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin.

–John von Neumann, 1951

We are going to implement a simple game that I learned to play recently. It is called *Left, Right, and Center*. It requires no skill, no real decision-making, and a player that is out of the game can suddenly come back in and often wins.

1 Playing the Game

Some number of players $1 < k \leq 10$ sit around a table, each player has in her hand \$3. There are three dice, and each die has 6 faces and is labeled: $3 \times \bullet$, $1 \times$ **L**, $1 \times$ **R** or $1 \times$ **C**.

1. Beginning with player 1, roll the dice:
   (a) If the player has \$3 or more then she rolls three dice; if she has \$2 then she rolls two dice; if she has only \$1 then she rolls one die; if she has no money then she must pass.
   (b) If the player rolls **L** then she gives \$1 for each **L** to the player on her *left*.
   (c) If the player rolls **R** then she gives \$1 for each **R** to the player on her *right*.
   (d) If the player rolls **C** then she puts \$1 for each **C** in the pot in the center.
   (e) If the player rolls • then she ignores it.
2. Move to the next player in sequence.
3. Repeat until only one player has any money.

Specifics

For grading purposes the numbering of the faces matters, and so they should be defined as follows (do not change it):

```
typedef enum faciem {LEFT, RIGHT, CENTER, PASS} faces;
faces die[] = {LEFT, RIGHT, CENTER, PASS, PASS, PASS};
```

You must give your players names, and for grading purposes the names should correspond to these (do not change them):

```
const char *names[] = {"Whoopi",
                       "Dale",
                       "Rosie",
                       "Jimmie",
                       "Barbara",
                       "Kyle",
                       "Raven",
                       "Tony",
                       "Jenny",
                       "Clint"};
```

You should think carefully about what quantities that you must track in order for your program to function. At a *minimum* you must keep track of the bank balance of each player, the amount of money in the pot, and the number of players that are *in*. Be careful: players that were *out* may be brought back in if money is passed to the *left* or *right*.

In lecture we talked briefly about *random* and *pseudo-random* numbers. As we know, computers produce pseudo-random numbers, and in this case it is to your benefit since *reproducibility* is essential. That means that in reality you program though it appears to be random is actually *deterministic*.

You accomplish rolling dice by calling the function `rand()` (read the *man page*) to get a random value from 0...5. This is used to determine what was rolled.

In order that your program be reproducible, you must start from a known place. This accomplished by *setting the random seed* using the function `srand()` (again read the *man page*).

Your program will ask for two numbers:

1. The random seed, and
2. The number of players.

The random seed completely determines the outcome of your program. If you give it the same random seed and the same number of players you *must* get the same answer.

Submission

You *must* turn in your assignment in the following manner:

1. Have file called `Makefile` that when the grader types `make` will compile your program. Since you have not learned about them yet in the laboratory section, here is one that you can use.

```
CFLAGS=-Wall -Wextra -Werror -pedantic
CC=gcc $(CFLAGS)

lrc     :       lrc.o
        $(CC) -o lrc lrc.o
lrc.o   :       lrc.c
        $(CC) -c lrc.c
clean   :
        rm -f lrc lrc.o
```

Makefile

2. A plain text file called `README` that describes how your program works.

3. The source file *must be called* `lrc.c`.
4. The executable file produced by the compiler *must be called* `lrc`.
5. These files must be in the directory `assignment0`.
6. You must `commit` and `push` the directory and its contents using `git`.

Example

```
unix [109]% ./lrc
Random seed: 1234
How many players? 4
Whoopi rolls... gives $1 to Jimmie gets a pass gets a pass
Dale rolls... gets a pass gets a pass gets a pass
Rosie rolls... gives $1 to Dale puts $1 in the pot puts $1 in the pot
Jimmie rolls... gives $1 to Rosie gets a pass gets a pass
Whoopi rolls... gets a pass gets a pass gives $1 to Jimmie
Dale rolls... gets a pass puts $1 in the pot gives $1 to Whoopi
Rosie rolls... gets a pass
Jimmie rolls... gets a pass puts $1 in the pot puts $1 in the pot
Whoopi rolls... puts $1 in the pot gives $1 to Jimmie
Dale rolls... gives $1 to Whoopi gives $1 to Whoopi
Rosie rolls... gets a pass gets a pass
Jimmie rolls... gives $1 to Rosie puts $1 in the pot puts $1 in the
    pot
Whoopi rolls... gives $1 to Jimmie
Rosie rolls... gives $1 to Dale gets a pass puts $1 in the pot
Jimmie rolls... gets a pass gives $1 to Rosie
Rosie rolls... gets a pass gets a pass
Jimmie rolls... gives $1 to Rosie
Rosie wins the $9 pot with $3 left in the bank!
```