Activity #2: The "Human Machine" Language

Here are the beginnings of a more formalized low-level language you can use to create programs for a "Human Machine" to solve problems with playing cards.

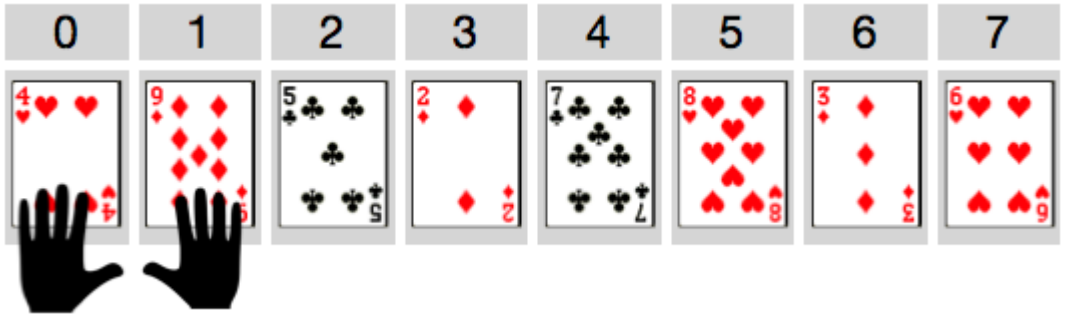
To simplify things we'll get rid of the need to pick cards up and put them down. Instead leave cards face up and just touch them. The 5 commands you can use are shown to the right. **See the *Reference Guide* on the next page for descriptions of what these commands do.**

```
SHIFT hand TO THE dir
MOVE hand TO POSITION num
JUMP TO LINE num
JUMP TO LINE num IF num comp? num
STOP
```

Some of these commands might seem unusual, but we can write programs with just these commands to control the "human machine's" hands to touch or pick up the cards, look at their values, and move left or right down the row of cards.

Standard Card Setup

You should assume this standard initial setup. Here is a diagram for an 8-card setup:



- There will be some number of cards with random values, lined up in a row, face up.
- Positions are numbered starting at 0 and increasing for however many cards there are.
- The left and right hands start at positions 0 and 1 respectively.

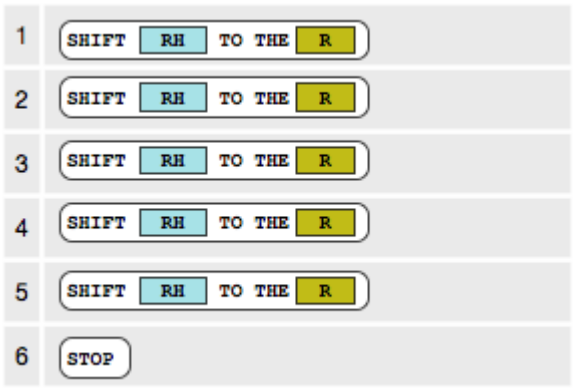
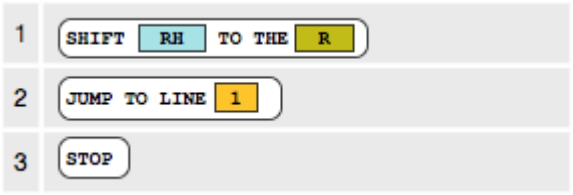
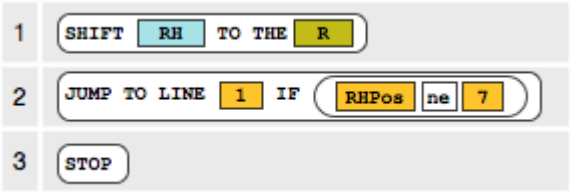
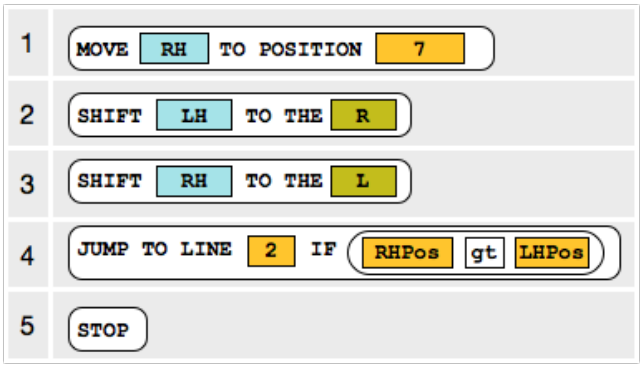
Try out some example programs

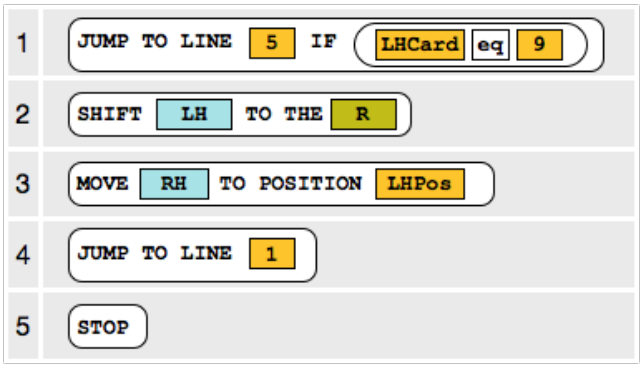
Get to know the Human Machine Language by acting out the examples on the following page with a partner. For each of the examples on the next page you should:

- Lay out a row of **8 cards** in front of you to test out the program.
- Have one partner read the instructions in sequence starting at line 1, and the other partner act out each command as the human machine.
- Use the ***code reference*** to answer your questions and verify you're interpreting the code correctly.
- Give a brief description of what the program does, or its ending state.

NOTES:

- Some of the programs are very simple
- Some of the programs might not ever stop
- The point is simply to practice using the language and executing commands as a "Human Machine"

Example Program	What does it do?
	
	*Note: this one has a problem, can you find it?*
	
	



Note: there is a potential problem with this one too. But only in certain circumstances. Can you find it?

Human Machine Code Reference Guide

Hands, Values and Direction

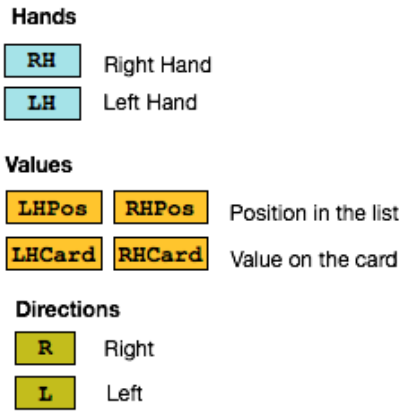
There are some short-hand abbreviations for referring to the human machine, the cards, positions, and directions of movement.

Hands - The Human Machine has hands! You can refer to a specific hand abbreviated **`LH`** or **`RH`** (left hand or right hand).

Values - Each hand has two values you can refer to:

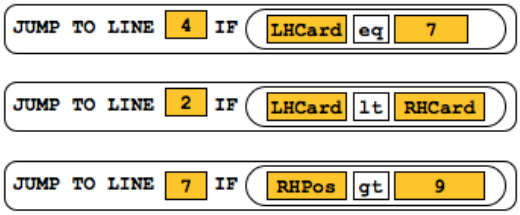
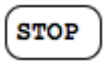
1. **`LHPos`**, **`RHPos`** - The hand's position in the list (a number)
2. **`LHCard`**, **`RHCard`** - The value of the card the hand is holding (a number)

Direction - There are two directions **`R`** and **`L`** (right and left) that hands can move along the row of cards.



Commands

Description	Examples
SHIFT *hand* TO THE *dir* Shift the given hand one position to the right or left along the row of cards.	SHIFT LH TO THE R
MOVE *hand* TO POSITION *num* Move a given hand to a specific position number in the row of cards.	MOVE RH TO POSITION 4 MOVE LH TO POSITION RHPos
JUMP TO LINE *num* Jump to a specific line number in the program and continue execution from that point.	JUMP TO LINE 1

 Jump to line but ONLY IF the comparison of two numbers is *true*. If the comparison is *false* then just proceed onto the next line of code. • For numbers, you can use **integers** or any of the hand values **`RHCard, LHCard, RHPos, LHPos`** • For comparisons you can use **`eq, ne, lt, gt`**, (equal, not equal, less than, greater than)	
 End of program. Stop doing anything, stop executing lines of code.	*This should be the last line of code in the program, or on a line that is jumped to when you want the program to stop.*