# Program #1 - Wheel of Decision

*Prof Bill - Mar 2018*

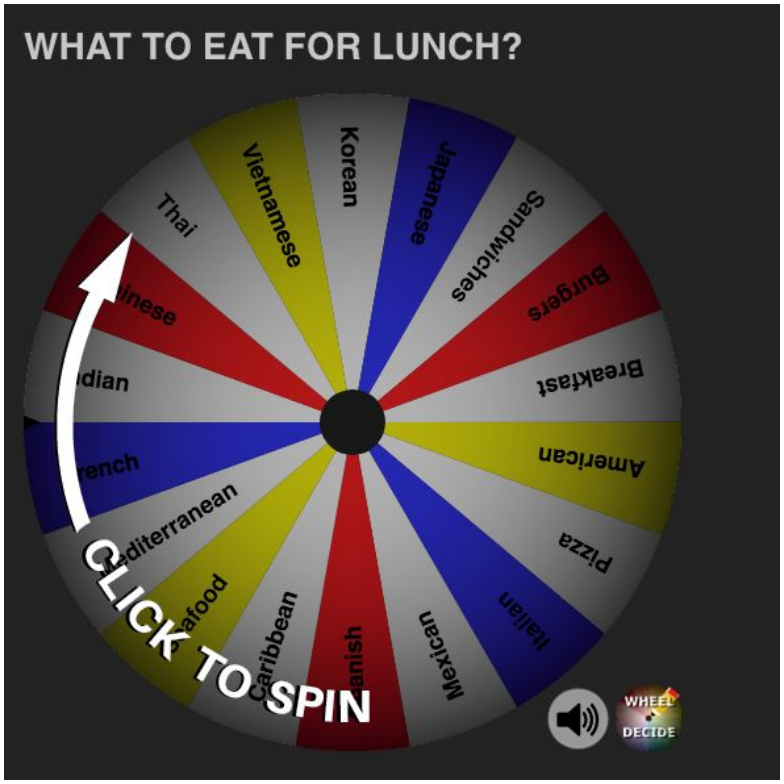
Program #1 logistics:

- Due: **Tue Apr 10, 2018** at the beginning of class (2 weeks)
- Worth: **8 points** (8% of your grade)
- Learn: linked lists, C programming language, debugger, pointers, patience

## 1. Description

Let's code up our own version of [wheeldecide.com](wheeldecide.com). It'll be handy to use the Wheel of Decision to randomly make those important career and relationship choices in our lives.

We'll write Program #1 in the C Programming Language and run it from the command line (text, no graphics). For more details, keep reading...
thanks… yow, bill



## 2. Commands

Your wheel program will run in the console. So, you'll type in your commands. Your wheel should support these commands.

| Command <args> | Description |
|---|---|
| `add <item>` | Add item to the wheel (end of the list) |
| `name <wheel>` | Name your wheel (optional, what is default?) |
| `print` | Print wheel items (in order) |
| `size` | Print number of items in wheel |
| `first` | Print first wheel item |
| `last` | Print last wheel item |
| `random` | Print a random wheel item |
| **`spin`** | **Spin the wheel, show special effects, select item at random, remove it, show item** |
| `reload` | Reload the wheel back so all items are available |
| `reverse` | Reverse the items in the wheel list |
| `clear` | Clear all items from the wheel |
| `read <file>` | Read a wheel file* |
| `save <file>` | Save your wheel to a file* |
| `exit` | Exit the program |

* The wheel file format will be simple: plain text, first line is the name, each remaining line in a wheel item. We'll put some fun ones on the k: drive once we get going.

## 3. Requirements

Program #1 requirements are:

- ➢ Write your program in **C**.
- ➢ Implement the **wheel commands** listed above.
- ➢ Code your own **doubly-linked list** with a head and tail.
- ➢ Flex **your creativity** by coming up with *at least* one own cool command or feature on your own.

## 4. Details

We'll work on some command sessions in class to work out the kinks on this interface. More on the operation of your wheel:

- Your "spin" should have some special effects related to it. Obviously, we can't show a spinning wheel, but we can print "spinning…" and add some delay. Your choice… get creative here.
- Commands with params are a little tricky. I recommend: Get the command, then ask for the parameter on a separate line. Then, you won't have to parse strings.
- Don't worry about elaborate error messages or checking. Do something logical. Don't crash and burn.
- Command shortcut? It looks like the first two letters are unique. Kust cayin.

Little help?

- See my C Programming helper on the class website.
- I'll post examples on the k: drive.
- We'll spend class time on this as well.

How to succeed (writing any program):

1. Start early!
2. Don't be shy. Ask a question in class. Email me. Come to office hours.
3. Small bites. Divide and conquer your program into small, manageable tasks.
4. ABW. Always be working. Your program should always compile and run. Never leave your work in disarray.

## 5. Grading

Create a **program1** folder on your k: drive.This folder should contain:

- All your C source files
- Your program1 executable
- Any test input and output files that you have
- A **README.txt** file where you describe the status of your program and the creative command that you added

All your code must follow our class **Coding Guidelines**. Ugly code will be severely penalized. A program that doesn't even compile is probably worth 0 points.

Remember our **plagiarism** guidelines as well. Getting help from google or stackoverflow or a friend is OK, but:

1. You must acknowledge any help you receive with a comment in your code
2. You must understand any code in your solution
3. Get help on program components, not the assignment (the tic tac toe philosophy)
4. If you have any questions in this area, contact me **before** you turn in your work, not after (when it's too late)

thanks… yow, bill

PS - Program #2?