## Midi Maze: Input

Input code should be in a separate C code file, usually called input.c. It might have only 1 function in it for now, but in future labs, other input routines will go here.

### Input files such as xA.mm

The xA.mm sample input file has two lines of data:

```
4107    2.0     20.0    120.0
4213    16.5    20.0    210.0
```

- The first token is a hexadecimal value to read into a C short. Your bits code (separate document) will deal with validating the code and extracting the things we want out of it.
- The initial X value is next.
- The Y value follows the initial X value.
- The direction, in degrees, is last.

The data is whitespace separated. Read all 4 in one attempt (use scanf and pay attention to the return value). There is no guarantee that all four items will be on one line and you should not care; whitespace separated is all you need to know.

### Input Processing

High level function in lab2.c will call the master input function in input.c. The input code needs to keep looping as long as scanf keeps reading cleanly. If scanf fails to read 4 things, the program should clean up and exit gracefully with a return value of 0, which is the same as EXIT_SUCCESS. In text mode it will also output the value returned by scanf. With every clean read check the validity of the code and if it is valid, launch a new simulation with the newly read data. If the code is not valid but 4 items were read, the program should abort, doing any required cleanup, and return the value 2 to the operating system.