Question 2 (11 marks)

In the `q2` directory (in the file `main.c`) is a program which

- reads words, one per line, from standard input
- normalises each word by converting it to all lower-case
- builds a trie using the normalised words as keys
- then prints all of the keys in the trie, one per line, on standard output

The overall effect is that the program prints a list of all the words in the input, in some order not necessarily related to the input order, and with all words being normalised.

The `q2` directory, along with the main program, contains an implementation of a trie ADT. The `Trie` type is defined in the files `Trie.h` and `Trie.c`. The `Trie` ADT contains the usual collection of functions (create a new `Trie`, insert a new `Item` containing a key, search for `Items` via their key values). It also contains a skeleton implementation of a function (`showKeys()`) to print all of the key values in the `Trie`.

Your task for this question is to implement the `showKeys()` function in the `Trie.c` file.

The `showKeys()` function is defined as follows:

```
void showKeys(Trie t)
```

It takes a `Trie` value (pointer to a `TrieRep` structure) as input, iterates over the `Trie` and writes each key to the standard output on a line by itself.

You can find out more about the behaviour of the `q2` program by looking at the files in `q2/tests` directory. Each file named `tX.sh` contains the commands to run one test. Each test will use one of the files named `tX.in` as input. Each test has a corresponding file `tX.exp` which contains the expected output from a correct implementation of `q2`, run using `tX.sh`.

The `q2` directory also contains a `Makefile` which you use as:

```
make q2    # build the q2 program
```

You can test your `q2` program using the command:

```
check q2    # run tests on the q2 program
```

Once you are satisfied with your program, submit it using the command:

```
submit q2
```

This will make a copy of the `Trie.c` file from the `q2` directory as your answer for this

question. You can run the `submit` command as many times as you like, but make sure that your final submission compiles without any errors or warnings. Test your program thoroughly, possibly using test cases additional to those supplied. Your program will be tested using inputs which are different to the examples in the `q2/tests` directory.

You can add any additional functions (apart from `showKeys()`) to the `Trie.c` file, but you may not change any of the other files.

If, at some stage, you need to "re-install" the files (although you should not need to), you can copy all of the original files into the `q2` directory by running the command:

```
re-start q2
```

Beware: this will overwrite all of your exsting files for this question, so only do it if you seriously mess things up.