

Exercises, Week 2, Day 4

1. Write an input function based on `raw_input` that will ask the user for an integer and - if the the input is not correct and one is not given - repeats its request.
2. Write a program that takes a name of a raw sequence file as an argument. It will process this file looking for sequencing errors or problems. These are defined as more than two unknowns ? in a row. Save the sequence with these regions replaced by three asterisks. `input2-4a.seq` can be regarded as a typical input.
3. Modify it so it can accept sequences that are spread over several lines and formatted as in `input2-4b.seq`. Note however that the output does not have to be formatted.
4. Modify the program so it replaces the error regions with the exact number of asterisks.
5. Write a parser for a phylip file. Descriptions can be found [here](#), [here](#) and [here](#). An example is given as `input2-4c.phylip` and you can assume this is typical. The contents should be returned as something logical (e.g. a list of name-sequence pairs). You can assume there are no errors in the file. Modules and functions like regular expressions, `split` and `replace` will be very useful.