JSON TOKENIZER

For this phase, your project should create the internal representation of JSON entity sets. That is, your project should parse the input tokens and build an instance of EntitySet from instances of EntityInstances, which in turn, consists of instances of Pair. Each of these classes should have a function called print, which prints the information that they hold. For example, print of Pair could look like this:

```
void Pair::print(int numSpaces, bool addComma);
```

When called, the function first prints "numSpaces" blank spaces, followed by the attribute name and value of the pair, and finally, if "addComma" is true, it prints a comma, on the same line, before the new-line character. Therefore, if the instance of Pair represents attribute name "term" whose value is 2193, the following call:

```
instanceOfPair.print(5, true);
```

Should produce:

```
     "term": "2193",
```

Note that the quotes are not stored in instances of Pair. They get added when the attribute name and attribute values are printed.

A call to print of EntityInstance would print the following for example:

```
{
    "gpa": 2.33256,
    "term": "2197",
    "id": "MDcwNTAwMDQxNDUzNTkwMjky"
}
```

Finally, EntitySet's print would print its data like this:

```
[
   {
      "gpa": 2.33256,
      "term": "2197",
      "id": "MDcwNTAwMDQxNDUzNTkwMjky"
   },
   {
      "gpa": 3.97562,
     "term": "2197",
```

```
      "id": "MDMwOTAxMTkwNTIxMzc5NzMx"
    }

]
```

As you notice, the idea is that, for this phase, you read a JSON file, build its internal representation, and print it out in the same format as the input itself. The output should get printed to the standard output stream.