Daily Notes COP 2930 – 11/17/2020

Two contexts for reading input from files:

1) Example of a contest question where we solve a problem and then run our solution on many test cases that are already stored in an input file.
2) More real life – get data from files, load into our program and then calculate something interesting.

Key Steps

1. Open
2. Read stuff from the file
3. Close

How to Read stuff – I will teach two ways:

1. Line by line
2. Token by token

To open a file to read from:

```
file = open(filename, “r”)
```

To read the next line from the file (with the \n) we do:

```
file.readline()
```

Generally we never call this on a line by itself. Instead we store what it returns (a string) somewhere.

If we know that string is in an integer, we can do:

```
num = int(file.readline())
```

If we want to store the whole line as a string, we do:

```
line = file.readline()
```

If we want to get rid of the \n at the end of the line we do:

```
line = file.readline().rstrip()
```

If we want to strip whitespace from both sides, we do line = file.readline.strip()

Reading by line is good if perhaps we have a name on a line and the name has spaces, commas, hyphens and stuff like that.

Reading to the end of file

Just read in the whole file at once into a list of strings, where each string is a line from the file:

lines = list(file)

Now you can index into the list lines to get whichever line you want.

Let's say the file stores:

```
Hello
Bob
Bye
This is another line.
```

Then list will look list:

```
lines = ["Hello\n", "Bob\n", "Bye\n", "This is another line.\n"]
```

Reading by token

```
If line = "19 4 7 8"
```

Then line.split() returns

```
["19", "4", "7", "8"]
```

Reading the whole file into a string and then tokenizing:

```
Wholefile = file.read()
```

To tokenize into a list of tokens do:

```
allToks = file.read().split()
```