Lab 4.1 (Deadline Friday 11 February 23:59)

- Upload your code to Einstein to have it verified.

Contact list

- Write a program called *contacts_041.py* that takes the name of a file of contacts as a single command line argument. Each line of the contacts file consists of a name and phone number.
- Your program must read each contact and store it in a dictionary. The dictionary's keys are names and the dictionary's values are the corresponding phone numbers.
- Once the dictionary has been constructed the program should read all lines from `stdin`. Each line consists of a single name.
- For each name the program should retrieve and print the corresponding phone number. If a name cannot be mapped to a phone number the program should print `No such contact`.
- For example, after constructing a dictionary from `contacts_input_00_041.txt` your program should produce the output below when reading from `contacts_stdin_00_041.txt`:

```
$ cat contacts_input_00_041.txt
Sue 085-6442378
Jimmy 086-1223277
Maggie 087-8822001
Amy 087-3240516
Wendy 086-9112645
Sean 085-3445123
```

```
$ cat contacts_stdin_00_041.txt
Jimmy
Sue
Sean
Gwen
Wendy
Tommy
Maggie
Amy
```

```
$ python3 contacts_041.py contacts_input_00_041.txt < contacts_stdin_00_041.txt
Name: Jimmy
Phone: 086-1223277
Name: Sue
Phone: 085-6442378
Name: Sean
Phone: 085-3445123
Name: Gwen
No such contact
Name: Wendy
Phone: 086-9112645
Name: Tommy
No such contact
Name: Maggie
Phone: 087-8822001
Name: Amy
Phone: 087-3240516
```

Fancy contact list

- Write a new version of the above program called `fancy_041.py`. It functions similarly except contact details consist of two items: a phone number and an email address.
- For example, after constructing a dictionary from `fancy_input_00_041.txt` your program should produce the output below when reading from `fancy_stdin_00_041.txt`:

```
$ cat fancy_input_00_041.txt
Sue 085-6442378 sue@eircom.net
Jimmy 086-1223277 james@apple.com
Maggie 087-8822001 maggie@microsoft.com
Amy 087-3240516 amy@rte.ie
Wendy 086-9112645 wendy@physics.dcu.ie
Sean 085-3445123 sean@tcd.ie
```

```
$ cat fancy_stdin_00_041.txt
Jimmy
Sue
Sean
Gwen
Wendy
Tommy
Maggie
Amy
```

```
$ python3 fancy_041.py fancy_input_00_041.txt < fancy_stdin_00_041.txt
Name: Jimmy
Phone: 086-1223277
Email: james@apple.com
Name: Sue
Phone: 085-6442378
Email: sue@eircom.net
Name: Sean
Phone: 085-3445123
Email: sean@tcd.ie
Name: Gwen
No such contact
Name: Wendy
Phone: 086-9112645
Email: wendy@physics.dcu.ie
Name: Tommy
No such contact
Name: Maggie
Phone: 087-8822001
Email: maggie@microsoft.com
Name: Amy
Phone: 087-3240516
Email: amy@rte.ie
```

Word frequencies

- Write a program called `words_041.py` that calculates the frequency of words in lines of text read from `stdin`.
- Your program must store the totals as values in a dictionary where the corresponding words are the keys.

- Once totals have been calculated the program should print all words in alphabetical order along with corresponding totals. For example:

```
$ cat words_stdin_00_041.txt
This is a test. And a test is this.
How many tests are required to prove something?
I wonder if there is something wrong with my code...?
I shouldn't worry. I'll sort it in the end.
```

```
$ python3 words_041.py < words_stdin_00_041.txt
a : 2
and : 1
are : 1
code : 1
end : 1
how : 1
i : 2
i'll : 1
if : 1
in : 1
is : 3
it : 1
many : 1
my : 1
prove : 1
required : 1
shouldn't : 1
something : 2
sort : 1
test : 2
tests : 1
the : 1
there : 1
this : 2
to : 1
with : 1
wonder : 1
worry : 1
wrong : 1
```

- Hints:

  1. Convert all words to lower case. `'A'` and `'a'` should not be counted as separate words.
  2. Remember to strip any surrounding punctuation from words. You may find `string.punctuation` useful for this task.
  3. Use the `sorted()` function for sorting.

Vowel frequencies

- Write a program called *`vowels_041.py`* that calculates the frequency of each of the vowels *a*, *e*, *i*, *o*, *u* in lines of text read from `stdin`.
- Your program must store the totals as values in a dictionary where the corresponding vowels are the keys.
- When run against gettysburg.txt your program should produce the following output (note the output must be neatly *tabulated* and values must be displayed in *decreasing* order):

```
$ python3 vowels_041.py < gettysburg.txt
e : 167
a : 105
o :  95
i :  69
u :  21
```

- Hints:

  1. Convert all words to lower case. `'A'` and `'a'` should not be counted as separate vowels.
  2. Printing the dictionary items in order of decreasing values is tricky. You will most likely have to study the documentation for the `sorted()` function.
  3. Note how the output must be neatly tabulated. Remember that to print an integer `x` to a given width `w` you can use something like: `print(f'{x:{w}}')`.