1. Perkovic 4.27, 1st ed.

4.27 Write a function `f_copy()` that takes as input two file names (as strings) and copies the content of the first file into the second.

```python
>>> f_copy('example.txt', 'output.txt')
>>> open('output.txt').read()
'The 3 lines in this file end with the new line character.'

There is a blank line above this line.'
```

2. Perkovic 4.29, 1st ed.

4.29 Write a function `stats()` that takes one input argument: the name of a text file. The function should print, on the screen, the number of lines, words, and characters in the file; your function should open the file only once.

```python
>>> stats('example.txt')
line count: 3
word count: 20
character count: 98
```
4. Fill in the blanks below in the following Python shell session. It is possible to do this blindly without thinking and see the results as they are displayed in the shell. However, it would be much more beneficial if you read section 6.1 first, and tried to predict in advance what would be displayed.

```python
>>> D = {}
>>> type(D)

>>> D['mary'] = 'lamb'
>>> 'lamb' in D

>>> for i in range(88, 91):
    D[chr(i)] = i

>>> D['Y']

>>> D['Y'] = D['Z'] - 1
>>> del(D['mary'])
>>> D
```
5. Write a function `createDictFromLists(LKeys, LValues)` that returns a dictionary with key:value pairs corresponding to the list objects in `LKeys` and `LValues`. For example, if `LKeys = ['a', 'b', 'c', 'd']` and `LValues = [97, 98, 99, 100]`, then `createDictFromLists(LKeys, LValues)` should return the dictionary `{'a':97, 'b':98, 'c':99, 'd':100}`. 
*Hint: Think how to index into your two lists to add one key:value pair to your dictionary, and then repeat for all pairs in a for-loop. Make sure to begin with an empty dictionary.*