Lab 2

This lab is a warmup: it will not be collected. It involves using a simple graphics package to draw pictures under program control.

Part (i) Write a Python class definition `Worksheet` for the simple spreadsheet objects (worksheets) described below. A worksheet is a two-dimensional arrangement of cells, each of which contains a simple value (string, integer or float). The columns are indexed using letters ('A', 'B', and so on– at most 26!) while the rows are indexed numerically *beginning at one.* Each cell has a *label* of the form "B3" consisting of its column letter and its row number.

	A	**B**	**C**	**D**	**E**
1	"Murphy"	"Tom"	45	67	54
2	"Kelly"	"John"	65	42	35
3	"Healy"	"Anne"	54	72	81
4	"O'Brien"	"Michael"	35	47	62
5	"Smith"	"Susan'	62	71	48

The initializing method takes two parameters: a name for the worksheet (string) and a number (integer) representing the number of columns. Initially the worksheet contains zero rows; rows can be appended later (see below). The class definition should make the dimensions of the worksheet `w` accessible using `w.max_row` and `w.max_column` (the number of rows and columns, respectively), and should include implementations for the following methods:

w.read_cell(label) Return the value in the cell denoted by the string 'label'. Ignore the possibility of there being no cell with that label.

w.write_cell(label, newval) Replace the value in the cell denoted by the string 'label' with the value of parameter 'newval'. Return the old value from that cell. Ignore the possibility of there being no cell with that label.

w.append(self, newvals) Append a new row at the end of this worksheet i.e. beneath all the existing rows. Parameter 'newvals' is a list and is optional. If 'newvals' is provided, use its values to populate the cells of the new row left to right. If 'newvals' is not provided, the cells in the new row are populated with the value None.

w.show() Print out the contents of the worksheet. Use the example at the bottom of the page as a guide for how the output should be formatted.

Part (ii) Write a Python function `add_totals(w)` that takes a `Worksheet` object containing student names and marks for various modules as above and that returns a fresh worksheet object containing a modified copy of the original with one extra column added to the right (Column 'F' below) that contains the total of the module marks for that student. The result when printed (using `w.print()`) should appear as follows. Note that your code may interact with a `Worksheet` object using only the methods and values described above.

```
        A       B       C       D       E       F
=============================================================
[ 1]  : Murphy  Tom     45      67      54      166
[ 2]  : Kelly   John    65      42      35      142
[ 3]  : Healy   Anne    54      72      81      207
[ 4]  : O'Brien Michael 35      47      62      144
[ 5]  : Smith   Susan   62      71      48      181
-------------------------------------------------------------
```