Assignments

- Write a program that creates a list containing tuples with elements in the multiplication table, i.e.,

```
[(1,1,1), (1,2,2),...(10,9,90),
                          (10,10,100)]
```

- Write a program that creates two list containing tuples with elements in the multiplication table – one for odd numbers and one for even numbers

```
[(1,1,1), (1,2,2),...(3,1,3), (3,2,6)...]
[(2,1,2), (2,2,4),...(4,1,4), (4,2,8)...]
```

Assignments

- Write a simple calculator that accepts the input in the Reverse Polish notation (i.e., the postfix notation)
  - the expressions are accepted as commandline arguments
  - e.g., **`pcalc.py 1 2 3 + +`**
    prints out **`6`**

Assignments

- Write a program that for a given string prints out number of occurrences of individual characters
  - e.g., **mississippi**

```
m: 1 times
i: 4 times
...
```

Assignment

- Implement selection sort
  - a function that takes a list of ints and sorts it by selection sort

- overview of selection sort
  - finds the smallest value in the whole array (0:n) and swaps it with the first item
  - finds the smallest value in the rest of the array (1:n) and swaps it with the second item
  - finds the smallest value in the rest of the array (2:n) and swaps it with the third item
  - etc. until we reach the end of the array

Assignment

- Implement heapsort (without help of heapq)
  - a function that takes a list of ints and sorts it by heapsort

- overview of heapsort
  - sorting using a heap
  - heap – binary tree where each node has smaller value than its children
  - heap is constructed directly in the array
  - children of the node i are 2*i+1 and 2*i+2

  - pseudocode

```
procedure heapsort(a, count)
   heapify(a, count)
   end = count - 1
   while end > 0 do
      swap(a[end], a[0])
      end = end - 1
      siftDown(a, 0, end)
```