Programming Exercise 4

Generators

a) Write a generator named **produce** that takes a string as input and processes it a character at a time. If the character is a decimal digit, **n**, say, then **n+1** repetitions of the *next character* (digit or not) are returned as a string (so a digit of 0-9 means that 1-10 instances of the next character are desired, respectively; you may assume that the input does not end with a digit character.) If the character is not a digit, it is returned as-is. The following code should work as indicated below.

```
p = produce('A2B5E3426FG0ZYW3210PQ89R')

for s in p: print(s,end='')

print()


Output:

   A BBB EEEEEE 4444 666 F G Z Y W 2222 00 P Q 999999999 R
```

b) Write a function, **consume,** that takes an active iterator from a generator like **produce** as input, receives each string yielded by the generator, and prints the results out in groups of 3, *as it receives them* (do not save the entire output string produced and then divide it into 3's). The following code should work:

```
consume(produce('A2B5E3426FG0ZYW3210PQ89R'))

Output:

ABB BEE EEE E44 446 66F GZY W22 220 0PQ 999 999 999 R
```

Note that an *iterator* is passed to **consume**.

Submit both generators in a file named **gen.py**.