You have a file that needs to be divided into `n` chunks. While it would be straightforward to split the file into equal-bytes sizes and then write those chunks to file, you cannot write any incomplete lines to the files. This means that all of the `n` files that you create must have **no truncated lines**. If a split of a certain byte-size would result in a truncated line, then you can back off and only write the previous complete line. You can save the rest of it for the next chunk.

You can download Metamorphosis, by Franz Kafka (https://storage.googleapis.com/class-notes-181217.appspot.com/pg5200.txt) as the sample text. The file is of size 139055 bytes. Splitting into three pieces gives the following files and their respective sizes:

size	filename
46310	pg5200.txt_000.txt
46334	pg5200.txt_001.txt
46411	pg5200.txt_002.txt

The last line of the `pg5200.txt_000.txt` is the following:

```
her, she hurried out again and even turned the key in the lock so
```

The last line of the `pg5200.txt_001.txt` is the following:

```
there.  He, fortunately, would usually see no more than the object
```

As a final hint, splitting the same file into eight parts gives the following:

size	filename
17321	pg5200.txt_000.txt
17376	pg5200.txt_001.txt
17409	pg5200.txt_002.txt
17354	pg5200.txt_003.txt
17445	pg5200.txt_004.txt
17332	pg5200.txt_005.txt
17381	pg5200.txt_006.txt
17437	pg5200.txt_007.txt

You should think about making your file sizes as uniform as possible (this not graded, however). Otherwise, for a very long file, the last file may be inordinately large, as compared to the others. Your algorithm should pass through the file **exactly** once. You should assume that you **cannot** read the entire file into memory at once. If possible, you also want to minimize how much you move the file pointer around in the file. You should ensure that your code produces the file sizes that are indicated for each of the cases shown above.

Here is the function signature:

```
def split_by_n(fname,n=3):
    '''
    Split files into sub files of near same size
    fname : Input file name
    n is the number of segments
    '''
```

- Hint: Use `wt` as the file write mode.
- The individual filenames should include the original filename (`fname`) and a number indicating the current file sequence number in the split. For example, if `pg5200.txt` is the original file then the 8th division should be named `pg5200.txt 007.txt` . Your code should strive to produce file sizes as