Program2.cc - Sample Input / Output

1   Program usage

```
This is Program2, a program to make concordances for text files.
$Id: program2.cc,v 0.5 1996/11/26 23:49:48 ralph Exp ralph $
Copyright (C) 1996 Ralph L. Meyer, Spotswood, NJ 08884.
Program2 is licensed to users only under the conditions stated in the
GNU public license that comes with the distribution of this program.
Program2 comes with ABSOLUTELY NO WARRANTY; for details see the GNU public
license that accompanies this program.  You are welcome to redistribute this
software under certain conditions; for details see the GNU public license.
Send BUG reports to: meyer@princeton.edu.


USAGE:
program2 [-p[q][n:num] or -l[q][n:num]] or -s[q][num]] filename [outfile]
filename is the usual /filepath/fname to the file for which you wish to make
a concordance.  -p, -l, and -s are switches by which you can tell the program
whether you wish to concordance words in the file by the page on which they
are found, by the line on which they are found, or by the stanza in which
they are found.  The program considers a page to be delimited by ASCII char
12 (decimal) - FF, the formfeed character.  Every formfeed character passed will
increment the page counter by 1 page.  To indicate stanzas, stanza numbers in
the file must have the form: number>. i.e.: 1>  for stanza 1, 2> for stanza 2,
and so forth.  Stanza numbers must precede the stanza they delimit.  Line
numbers are incremented by the Line feed  character, ASCII character 10
(decimal).  Default operation (no switches indicated on the command line)
is to indicate the page number in the file on which a word will be found.
Note that only one page/line/stanza switch at a time is allowed!
The addition n:number can be appended to any of the type-of-count switches
and the beginning number of the page or line will be changed to the number
following the n: (n:num has no effect on stanza counting).  If an outfilename
is used, there MUST be a switch indicating the count type on the command line.
The auxiliary switch 'q' causes only the copyright to be output to stdout.
Without 'q' program2 outputs information on the filenames in which it has
stored results.
```

```
Example:

program2 -pqn:40 MY.TXT MYCORD [Enter]

...would cause the text file MY.TXT to be concordanced by pages with the first
page considered to be page 40. No messages would be output to screen but the
opening copyright notice.  Output would be put in the files MYCORD.wrd and
MYCORD.abc .  Program2 creates 2 files, one named 'filename.wds', which
contains a concordance of all the words in the file 'filename'.  The words are
listed alphabetically.  The number preceding the word is the word's length.
the number immediately following the word indicates the number of times
that word was used in the document filename.  The succeeding numbers
indicate the page/line/stanza numbers in which the word was found.


In addition to producing the file 'filename.wds', program2 also outputs
a document named 'filename.abc' that contains a listing of the letters
of the English alphabet, and the numerals 0-9, whose usage in the document
'filename.ext' has been counted and graphed.

After running program2, look for these two files named for the file that was
concordanced or the name you gave the outfile.  They may be examined using
the command 'cat', 'less', or by using your favorite text editor.

Please Note that very long documents may need to be separated into pieces
to be concordanced.  Larger documents should be broken into smaller pieces,
```

2   input file

```
input
 1 This is an input file.
 2 How's going?
 3 Great! I got an A+ in SE class.
 4 That's awesome. You deserve it.
 5 Thank you!
```

3.1 output1

```
output.wds
 1 Concordance for file: input
 2
 3 Note: Concordancer considers a word a set of alphanumeric characters
 4       bounded by whitespace at each end of the set.
 5
 6 Total number of words in file: 22
 7 Initial page of file was page 1
 8 Last page of file was page 1
 9
10 Word              Number of    Page number
11 Len  Word:            uses:    Locations:
12   1 a...................1 - 1
13   2 .............an.....2 - 1
14   7 ........awesome.....1 - 1
15   5 ..........class.....1 - 1
16   7 ........deserve.....1 - 1
17   4 ..........file.....1 - 1
18   5 ..........going.....1 - 1
19   3 ...........got.....1 - 1
20   5 .........great.....1 - 1
21   5 .........how's.....1 - 1
22   1 ..............i.....1 - 1
23   2 .............in.....1 - 1
24   5 .........input.....1 - 1
25   2 .............is.....1 - 1
26   2 .............it.....1 - 1
27   2 .............se.....1 - 1
28   5 ..........thank.....1 - 1
29   6 .........that's.....1 - 1
30   4 ..........this.....1 - 1
31   3 ...........you.....2 - 1
```

3.2 output2

```
output.abc
 1  a ....8     10% aaaaaaaaaa
 2  b ....0      0%
 3  c ....1    1.3% cc
 4  d ....1    1.3% dd
 5  e ....8     10% eeeeeeeeee
 6  f ....1    1.3% ff
 7  g ....4    5.1% gggggg
 8  h ....4    5.1% hhhhh
 9  i ....8     10% iiiiiiiiii
10  j ....0      0%
11  k ....1    1.3% kk
12  l ....2    2.5% lll
13  m ....1    1.3% mm
14  n ....6    7.6% nnnnnnnn
15  o ....6    7.6% oooooooo
16  p ....1    1.3% pp
17  q ....0      0%
18  r ....2    2.5% rrr
19  s ....9     11% sssssssssss
20  t ....8     10% tttttttttt
21  u ....3    3.8% uuuu
22  v ....1    1.3% vv
23  w ....2    2.5% www
24  x ....0      0%
25  y ....2    2.5% yyy
26  z ....0      0%
27  0 ....0      0%
28  1 ....0      0%
29  2 ....0      0%
30  3 ....0      0%
31  4 ....0      0%
32  5 ....0      0%
33  6 ....0      0%
34  7 ....0      0%
35  8 ....0      0%
36  9 ....0      0%
37
38  Total Characters: 79
```