Pascal Problem

Introduction: The soundex algorithm is often used to map surnames to codewords of the form 'Addd', where 'A' designates an alphabetic character and 'd' denotes a digit in the range 0..6. The goal of the algorithm is to map similar sounding surnames to the same codeword. For example, Smith and Smythe map to S530. The soundex technique is used to help resolve names in settings where misspellings or alternate spellings occur frequently. For example, it is used in genealogical research when attempting to locate all members of a family, some of whom may have spelled their last names differently from others.

Soundex algorithm: The following description is adapted from *The Art of Computer Programming, Vol. 3, Sorting and Searching*, by D. E. Knuth.

1. Retain the first letter of the name, and drop all occurrences of a, e, h, i, o, u, w, y in other positions.

2. Assing the following numbers to the remaining letters after the first:

   ```
   1: b, f, p, v
   2: c, g, j, k, q, s, x, z
   3: d, t
   4: l
   5: m, n
   6: r
   ```

3. If two or more letters with the same code were adjacent in the <u>original</u> name (before step 1), omit all but the first.

4. Convert to the form 'letter, digit, digit, digit' by adding trailing zeros (if there are fewer than three digits), or by dropping rightmost digits (if there are more than three).

Problem: Write an interactive program which reads a sequence of names, one name per line, from the keyboard, and produces a two-part output. The name 'ZZZZ' should be typed by the user to indicate end of input. You may assume that the names contain no blanks, but you may not assume that all characters are alphabetic (e.g., O'Brien, Sackville-West). (If a non-alphabetic character appears within the name, it should be ignored.) The algorithm (and thus your program) should not make a distinction between upper case and lower case letters. The first part of the output will be a table of names and corresponding soundex codes, printed as the names are entered and the codes computed. The second part of the output will be a frequency table summary of the soundex codes generated during this run of the program. In the summary, we want an ordered list of each letter and first digit combination which occurred during this run and how many codes began with that letter and first digit. (The list should be in ascending order by soundex code.)

Sample run of the program:

On the screen:

```
Welcome to the Soundex Program.
Enter a name ('ZZZZ' when done): Smith
  Soundex code for Smith: S530
Enter a name ('ZZZZ' when done): Harrison
  Soundex code for Harrison: H625
Enter a name ('ZZZZ' when done): Sturges
  Soundex code for Sturges: S362
Enter a name ('ZZZZ' when done): Smith-Grosvenor
  Soundex code for Smith-Grosvenor: S532
Enter a name ('ZZZZ' when done): ZZZZ
Thank you for using the Soundex Program.
```

Printed output:

```
Name             Soundex code
----             ------------
Smith              S530
Harrison           H625
Sturges            S362
Smith-Grosvenor    S532


       S U M M A R Y
Soundex Codes    Frequency
-------------    ---------
   H6--              1
   S3--              1
   S5--              2
```

Required output: Hand in a listing of your program together with the printed output generated for the following list of names:

```
Smith
Knuth
Baudot
Hickson
Mallory
O'Brien
Thackeray
Boylston
Boylson
Balston
Mennon
Smithson
Kant
deMolay
Howe
Lloyd
```

Hint: You might want to arrange your program so that you can easily suppress the output to the printer (or direct it to the screen instead); when you are debugging the program, you will presumably not want to be accumulating lots of printed output. Similarly, you might make provision for getting input from a text file, so that the above list of names does not have to be typed many times. (Also, if you don't do this, your printed output may get intermixed with someone else's.)