String Applications

Eric Roberts
CSCI 121
October 4, 2019

Augusta Ada Byron, Lady Lovelace (1815–1852)

• Augusta Ada Byron, daughter of the English poet Lord Byron, was encouraged to pursue her interests in mathematics before women were allowed to study at university.
• At the age of 17, Ada met Charles Babbage and became a collaborator on the design of his machines. Her work on the Analytical Engine has led many people to characterize her as the first programmer.
• More than Babbage, Ada saw that the Analytical Engine could store nonmathematical data, which is the foundation of string processing.

Generating Acronyms

• An acronym is a word formed by taking the first letter of each word in a sequence, as in
  "port out starboard home" — "posh"
  "not in my back yard" — "nimby"
  "self-contained underwater breathing apparatus" — "scuba"

• The text describes and implements two versions of a function acronym(s) that generates an acronym for s:
  - The first version searches for spaces in the string and includes the following character in the acronym. This version, however, fails for acronyms like scuba, in which some of the words are separated by hyphens rather than spaces.
  - The second version looks at every character and keeps track of whether the algorithm is scanning a word formed composed of sequential letters. This version correctly handles scuba as well as strings that have leading, trailing, or multiple spaces.

Improved acronym Function

Section 6.5 works through the design and implementation of a program to convert a sentence from English to Pig Latin. In this dialect, the Pig Latin version of a word is formed by applying the following rules:

1. If the word begins with a consonant, the wordToPigLatin function moves the initial consonant string to the end of the word and then adds the suffix ay, as follows:

   * scram — arecsam

2. If the word begins with a vowel, wordToPigLatin generates the Pig Latin version simply by adding the suffix way, like this:

   * apple — appleway

3. If the word contains no vowels at all, wordToPigLatin returns the original word unchanged.
Pseudocode for the Pig Latin Program

```python
def toPigLatin(str):
    result = []
    for char in str:
        if char.isalpha():
            if not result:
                result.append(char)
            else:
                for i, char in enumerate(result):
                    if char in 'aeiouAEIOU':
                        result = [char] + result[:i] + ['way'] + result[i:]
                        result += ['ay']
                        break
                else:
                    result.append(char)
        result += ['ay']
    return ''.join(result)

def wordToPigLatin(word):
    if not any(char in 'aeiouAEIOU' for char in word):
        return word
    else:
        vowels = [char for char in word if char in 'aeiouAEIOU']
        head, tail = word[:vowels[0]], word[vowels[0]:]
        return tail + head + 'ay'
```

Simulating the PigLatin Program

```python
toPigLatin('this is pig latin')
```

The `english.py` Library

- For Milestones #2 and #3 of the SpellingBee project, you need to use the `english` module, which is included in the starter project. Although the implementation of that module doesn't appear in the book until Chapter 8, you can certainly use it without understanding how it works in detail.
- The `english` module exports two resources:
  1. The constant `ENGLISH_WORDS`, which contains all the English words in alphabetical order.
  2. A predicate function `isEnglishWord`, which takes a string and returns `True` if that string is in the list of English words.
- The first example in Chapter 8 that uses the `english` module is a program called `TwoLetterWords`, which lists the legal two-letter words in the dictionary. The next slide animates a minor revision of that program that displays the words on a single line, which is then folded by the console.

The `TwoLetterWords` Program

```python
from english import ENGLISH_WORDS

def TwoLetterWords():
    for word in ENGLISH_WORDS:
        if len(word) == 2:
            print(word, end=' ')

TwoLetterWords()
```

Exercise: Finding “S” Hooks

- In Scrabble, one of the most important strategic principles is to conserve your S tiles so that you can hook high-scoring plays onto existing words.
- Some years ago, I was in a hotel where the shower taps were prominently labeled with HOT and COLD:

```
HOT
COLD
```
- Being a Scrabble player, it occurred to me that both of these words take an S on either end, making them ideally flexible for Scrabble plays.
- Write a Python program that uses the `english` module to find all such words.

Exercise: English Words in Pig Latin

- When you convert English to Pig Latin, most words turn into something that sounds vaguely Latinate but different from conventional English.
- There are, however, a few words whose Pig Latin equivalents just happen to be English words (not counting words without any vowels, which remain unchanged). For example, the Pig Latin translation of trash is ashtray, and the translation for express is expressway.
- Write a Python program that uses the `english` and `PigLatin` modules to find the 27 words that have this property.