Dmitri Stanchevici
Unit 3
Assignment 2

---

A 2.1

1. In main, find whether first vowel in word is 'a' and if yes, mark its position. If no, finish program with false.
2. From main, send to identifyOrder() an array of chars from word, an array of vowels in alphabetical order, position of 'a' in word, and index of 'a' in vowels (0).
3. In identifyOrder, use recursion to identify whether each subsequent vowel in word equals next vowel in vowels. Keep track of the indices of these vowels. If the next vowel in word does not correspond to the next vowel in vowels, return false, which cascades back to return false to main. If all vowels correspond in order (and all vowels are present in word), return true cascading back thru the previous recursions to return true to main.

---

A2.2

**Method:** checkOrder (word, positionInWord, vowels, positionInVowels)
**Input:** word and vowels as arrays
positionInWord: where to start looking for the next vowel
positionInVowels: location of next vowe

```
1.      // 1. Base case
        if positionInVowels >= vowels.length
                return true

        // 2. Base case
        if positionInWord >= word.length AND positionInVowels < vowels.length - 1
                return false

2.      c = vowels[positionInVowels]
3.      for k=positionInWord to word.length
4.              if word[k] == c
                // Recurse to find the next vowel from k+1 onwards.
5.                      checkOrder (word, k+1, vowels, positionInVowels+1)
6.              endif
```

7. **endfor**

---

# Problem 5

## A. Richard Stallman and Linus Torvalds

A quick survey of relevant Wikipedia articles and some other sources has revealed that Richard Stallman is a programmer and an important contributor to the philosophy and practice of software development. Specifically, Stallman is a vigorous promoter of free software, that is software resulting from the collaboration of numerous developers and that is open to free (in the sense of unlimited) modification and redistribution. Stallman has formalized the principles of free software movement in a document called “Free Software Definition” (https://www.gnu.org/philosophy/free-sw.html.en). Stallman is the founder of the GNU Project, a computer platform composed entirely of free software. The EMACS text editor that some Gateway students use is a GNU product.

The principles of free reuse codified in the GNU Project allowed Linus Torvalds, a Finnish-American programmer, to develop in the early 1990s the Linux kernel that underlies the widely used Linux operating systems. He also created Git, the popular open-source version control software. He espouses open-source software, which was perceived as opposed to proprietary software, because it made its source code open to all users, allowing them to modify and customize it as needed.  Torvald believed that such an approach benefited both users, because they have free access to collaboratively produced software, and the products, because they become more robust and agile due to continuous contributions by multiple developers. Torvalds has received multiple awards for contributions to the software industry.

## B. Open-Source Software

Both “open-source software” and “free software” describe software that results from mass collaboration, that is from continuous contributions of users and developers who use, study, share, and modify this software. Consequently these names (as well as such names as “free open source”) have been used interchangeably. However, Richard Stallman has objected to equating open-source software and free software, noting that the former is a development model while the latter is a philosophy nurturing attitudes to software as the product of free collaboration and thus belonging to all. Some critics, such as James Gosling, the creator of the Java programming language, find Stallman’s insistence on “free” (in the sense of gratis, even though the principle of unconstrained use and distribution is more important to Stallman) access

to software--and by extension to all creative products--to be extreme, depriving developers of their well-deserved rewards (https://www.youtube.com/watch?v=54RrHztFjP8).

This criticism was shared by commercial software developers, such as Microsoft. However, eventually such companies have adopted some open-software principles, freely publishing the source code of some of their products and sponsoring open-source software communities and projects.

Open-source software has made vital contributions to the computing world. In addition to the already mentioned Linux and Git, it lay at the foundation of the Internet. In 1999, an Open Source Initiative was established to promote open-source collaboration, articulate its ground principles, and authorize its licenses.

# C. Software Licensing; GPL and BDS

According to Lutkevich, “A software license is a document that provides legally binding guidelines for the use and distribution of software” (https://www.techtarget.com/searchcio/definition/software-license). Such a document is a contract between the developer, distributor, and end user of a piece of software, specifying each party’s rights and obligations in relation to the software. Software licenses are broadly categorized into proprietary and free-and-open-source. Proprietary licenses restrict end users’ access to the source code and prohibits unlicensed copying of the software. Such licenses allow developers to sell copies of their software and often obligates them to maintain the software during a specified period of time.

The initial impetus behind open-source software is a free and creative collaboration for improving software for all users. However, open-source software licenses vary in how much freedom they grant developers and users. The GPL General Public Licence, written by Richard Stallman, allows users to use, study, modify, and redistribute software, but obligates users/developers to make their products publicly available under the same terms as the ones under which they accessed the licensed software that served as the foundation for new products. Such terms of reciprocity are summarized in the name “copyleft.” Copyleft disallows the imposition of copyright of derivative products and prevents any restrictions on collaborative software development and distribution. Thus copyleft seems to be incompatible with monetary profit. The Java programming language was initially released under a proprietary license, but later it was re-licensed under a GPL license (https://en.wikipedia.org/wiki/Java_(programming_language)).

However, not all open-source licenses prevent charging fees for licensed products. Another class of open-source software does allow the imposition of copyright on derivative products and thus profiting from selling these products. Such licenses are often described as “permissive,” as opposed to the “protective” copyleft licenses (https://www.ibm.com/topics/open-source). This class includes BSD (Berkley Source Distribution) licenses. The Phyton programming language is managed under a BSD-style (a permissive) license. It is “compatible with the GNU General

Public License.” However, “Since the license is permissive, it allows proprietization of the derivations” (https://en.wikipedia.org/wiki/Python_Software_Foundation_License).