Programação I

Licenciatura em Engenharia Informática

2015–2016

Vitor Beires Nogueira

Escola de Ciências e Tecnologia
Universidade de Évora

Lendo listas de palavras

Exemplo (Abertura de um ficheiro para leitura)

```
>>> fin = open('words.txt')
>>> print(fin)
<open file 'words.txt', mode 'r' at 0xb7f4b380>
```

Exemplo (Lendo)

```
>>> fin.readline()
'aa\r\n'


>>> fin.readline()
'aah\r\n'
```

Leituras mais elaborada

Exemplo (Lendo)

```
>>> line = fin.readline()
>>> word = line.strip()
>>> print(word)
aahed


fin = open('words.txt')
for line in fin:
    word = line.strip()
    print(word)
fin.close()
```

Palavras sem a letra e

Escreva uma função `has_no_e` que devolve `True` se a palavra dada não tem a letra **e** (devolvendo `False` de outro modo).

```
def has_no_e(word):
    for letter in word:
        if letter == 'e':
            return False
    return True
```

Evitando determinadas letras

Escreva uma função **avoid** que recebe uma palavra e uma string de letras proibidas e devolve `True` se a palavra não tem nenhuma das letras e devolve `False` de outro modo.

```
def avoids(word, forbidden):
    for letter in word:
        if letter in forbidden:
            return False
    return True
```

Utilizando somente determinadas letras

Escreva uma função **uses_only** que recebe uma palavra e uma string de letras e devolve `True` se a palavra só contém letras da lista e devolve `False` de outro modo.

```
def uses_only(word, required):
    for letter in word:
        if letter not in required:
            return False
    return True
```

Ciclos com os índices

Escreva uma função **is_abecedarian** que devolve `True` se as letras de uma palavra que recebe aparecem por ordem alfabética (são permitidas letras duplicadas).

Exemplo (Sem índices)

```
def is_abecedarian(word):
    previous = word[0]
    for c in word:
        if c < previous:
            return False
        previous = c
    return True
```

Com recursividade

Exemplo (Com recursividade)

```
def is_abecedarian(word):
    if len(word) <= 1:
        return True
    if word[0] > word[1]:
        return False
    return is_abecedarian(word[1:])
```

Com um ciclo while

Exemplo (Com um ciclo while)

```
def is_abecedarian(word):
    i = 0
    while i < len(word)-1:
        if word[i+1] < word[i]:
            return False
        i = i+1
    return True
```

Palindrome

Escreva uma função `is_palindrome` que devolve `True` se a palavra que recebe é um *palindrome* (capicua).

Exemplo

```
def is_palindrome(word):
    i = 0
    j = len(word)-1

    while i<j:
        if word[i] != word[j]:
            return False
        i = i+1
        j = j-1

    return True
```

Debugging

Que exemplos devemos experimentar para testar a função `has_no_e`?

- palavras sem a letra *e*
- palavras com a letra *e*
  - no início
  - no meio
  - no fim
- palavras "compridas"
- palavras "muito curtas"
  - String vazia

Program testing

Program testing can be used to show the presence of bugs, but never to show their absence!
— Edsger W. Dijkstra