TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 6ª feira 12:30-14:00	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva uma função, `conta_duplicacoes`, que recebe uma cadeia de caracteres, correspondente ao nome de um ficheiro, lê esse ficheiro, linha a linha, e devolve um tuplo, cujo primeiro elemento é o número de linhas do ficheiro e segundo elemento é um dicionário cujos índices são caracteres e cujo valor corresponde ao número de vezes o esse caracter apareceu duas vezes seguidas no ficheiro. Por exemplo, se o ficheiro `cdex.txt` contiver

```
hggsf rr sftra
hhsr
pooooyy rtt
Hfddsa
```

então

```
>>> conta_duplicacoes('cdex.txt')
(4, {'d': 1, 'g': 1, 'h': 1, 'o': 3, 'r': 1, 't': 1, 'y': 1})
```

Solução:

```
def conta_duplicacoes(nome):
    f = open(nome, 'r')

    n_linhas = 0
    res = {}
    for linha in f:
        n_linhas += 1
        for c in range(1, len(linha)):
            if linha[c] == linha[c - 1]:
                res[linha[c]] = res.get(linha[c], 0) + 1

    f.close()
    return(n_linhas, res)
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 6ª feira 14:00-15:30	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva a função `cria_f_index` que recebe uma cadeia de caracteres correspondente a um ficheiro de entrada e devolve um dicionário cujas chaves correspondem ao primeiro caracter das linhas e cujo valor corresponde ao número de linhas que começam por um determinado caractere. Por exemplo, se o ficheiro texto.txt tiver:

```
Isto é um exemplo
De um ficheiro
Onde existem 4 linhas
De palavras sem nexo
```

É produzida a seguinte interacção:

```
>>> cria_f_index('texto.txt')
  {'O': 1, 'I': 1, 'D': 2}
```

Solução:

```
def cria_f_index(file):
    f = open(file, 'r', encoding='UTF-8')
    lines = f.readlines()
    f.close()
    d = {}
    for line in lines:
       if not line[0] in d:
            d[line[0]] = 0

       d[line[0]] = d[line[0]] + 1
    return d
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 6ª feira 15:30-17:00	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva a função `cria_f_index` que recebe uma cadeia de caracteres correspondente ao nome de um ficheiro de entrada e devolve um dicionário cujas chaves correspondem ao primeiro caracter das linhas e cujo valor corresponde ao número de linhas que começam por um determinado caractere. Por exemplo, se o ficheiro `texto.txt` tiver:

```
Isto é um exemplo
De um ficheiro
Onde existem 4 linhas
De palavras sem nexo
```

É produzida a seguinte interacção:

```
>>> cria_f_index('texto.txt')
  {'O': 1, 'I': 1, 'D': 2}
```

Solução:

```
def cria_f_index(file):
    f = open(file, 'r', encoding='UTF-8')
    lines = f.readlines()
    f.close()
    d = {}
    for line in lines:
       if not line[0] in d:
            d[line[0]] = 0

       d[line[0]] = d[line[0]] + 1
    return d
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 2ª feira 09:00-10:30	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva a função `file_column` que recebe uma cadeia de caracteres correspondente ao nome de um ficheiro de entrada, um inteiro correspondente à coluna e um caráter que funciona como separador. A função deve imprimir no terminal o texto de cada linha correspondente à coluna indicada, considerado que as colunas são divididas pelo separador. Considere que a primeira coluna é a 0 (zero). Num ficheiro .csv pode fazer `file_colunm('texto.csv', 3, ':')` para obter a terceira coluna da tabela. Por exemplo, se o ficheiro `texto.txt` tiver:

```
Isto é um exemplo
De um ficheiro
Onde existem 4 linhas
De palavras sem nexo
```

É produzida a seguinte interacção:

```
>>> file_colunm('texto.txt', 1, ' ')
é
um
existem
palavras
```

Solução:

```
def file_column(file_in, column, sep) :
  fin = open(file_in, 'r')
  lns = fin.readlines()
  fin.close()
  for ln in lns :
    col, out = 0, ''
    for ch in ln :
      if ch == sep :
        col += 1
      elif col == column :
        out += ch
    if out != '' :
      print(out)
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 2ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 10 - Ficheiros

Escreva a função `word_count` que recebe uma cadeia de caracteres correspondente ao nome de um ficheiro de entrada e devolve um tuplo com três inteiros correspondentes ao número de linhas no ficheiro, ao número de palavras (separadas por espaços) e número de carateres. Por exemplo, se o ficheiro `texto.txt` tiver:

```
Isto é um exemplo
De um ficheiro
Onde existem 4 linhas
De palavras sem nexo
```

É produzida a seguinte interacção:

```
>>> word_count('texto.txt')
( 4, 15, 76 )
```

Solução:

```
def word_count(file_in) :
  chars = 0
  words, instr = 0, False
  fin = open(file_in, 'r')
  lns = fin.readlines()
  fin.close()
  for ln in lns :
    chars += len(ln)
    for ch in ln :
      if ch == ' ' or ch == '\n' :
        if instr :
          instr = False
      else :
        if not instr :
          instr = True
          words += 1
  return (len(lns), words, chars)
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 5ª feira 10:30-12:00	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva a função `equal_crc` que recebe uma cadeia de caracteres correspondente ao nome de um ficheiro de entrada base e uma lista de cadeias de caracteres correspondente aos nomes de outros ficheiros. A função devolve um tuplo de nomes de ficheiros com o mesmo `crc` que o ficheiro de entrada base. Considere a rotina já existente:

```
def crc(lines) :
  code = 0
  for ln in lines :
    for ch in ln :
      code = ( code * 11 + ord(ch) ) % 1000000
  return code
```

É produzida a seguinte interacção:

```
>>> equal_crc('ex.txt', [ 'f1.jpg', 'f2.csv', 'f3.zip' ] )
( 'f1.jpg', 'f3.zip' )
```

Solução:

```
def equal_crc(file_in, lst) :
  out = ()
  fin = open(file_in, 'r')
  lns = fin.readlines()
  fin.close()
  crc_in = crc(lns)
  for fname in lst :
    fin = open(fname, 'r')
    lns = fin.readlines()
    fin.close()
    if crc_in == crc(lns) :
      out += fname,
  return out
```

TÉCNICO LISBOA

Fundamentos de Programação - 2019/2020 Aula Prática 11 (30 minutos) Turno 4ª feira 8:00-9:30	
Nome:	
Número:	
Data:	
Curso:	

Capítulo 10 - Ficheiros

Escreva a função `wrap_lines` que recebe duas cadeias de caracteres correspondentes ao nome de dois ficheiros, de entrada e de saída respectivamente, e um número inteiro correspondente ao maior número de colunas a produzir no ficheiros de saída. A função deve copiar as linhas do ficheiro de entrada para o ficheiro de saída, garantindo que a dimensão máxima de colunas indicada nunca é excedida. Por exemplo, se o ficheiro `texto.txt` tiver:

```
Isto é um exemplo
De um ficheiro
Onde existem 4 linhas
De palavras sem nexo
```

É produzida a seguinte interacção:

```
>>> wrap_lines('texto.txt', '/dev/tty', 15)
Isto é um exemp
lo
De um ficheiro
Onde existem 4
linhas
De palavras sem
 nexo
```

Solução:

```
def wrap_lines(file_in, file_out, col) :
  fin = open(file_in, 'r')
  lns = fin.readlines()
  fin.close()
  out = open(file_out, 'w')
  for ln in lns :
    while len(ln) > col :
      out.write(ln[:col]+'\n')
      ln = ln[col:]
    if len(ln) > 1 :
      out.write(ln)
  out.close()
```