Processamento/Teoria de Linguagens e Compilação
LCC (3ºano) + MEFis (1ºano)

Trabalho Prático nº 1 (ER + Filtros de Texto)

Ano lectivo 21/22

Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- aumentar a capacidade de escrever *Expressões Regulares (ER)* para descrição de *padrões de frases* dentro de textos;
- desenvolver, a partir de ER, sistematicamente *Processadores de Linguagens Regulares*, ou *Filtros de Texto (FT)*, que filtrem ou transformem textos com base no conceito de regras de produção *Condição-Ação*;
- utilizar o módulo 're'—com suas funções de search(), split(), sub()—do Python para implementar os FT pedidos.

Para o efeito, esta folha contém 5 enunciados, dos quais deverá resolver um escolhido em função do número do grupo (NGr) usando a fórmula $exe = (NGr \ \% \ 5) \ + \ 1$.
Neste TP que se pretende que seja resolvido rapidamente), aprecia-se a imaginação/criatividade dos grupos ao incluir outros processamentos!

O ficheiro único com o relatório e a solução deve ter o nome '`plc21TP1grNGr`' e será submetido até ao fim da data indicada através do Bb.
O programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data a marcar.
O **relatório** a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da solução e sua implementação (incluir o código Python), deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido).
Como é de tradição, o relatório será escrito em LaTeX.

1 Processador de Inscritos numa atividade desportiva

Construa agora um ou vários programas Python para processar o texto '`inscritos.txt`' conforme solicitado nas alíneas seguintes:

a) imprimir o nome e o email dos concorrentes inscritos entre a 5º e a 15º posições.

b) imprimir o nome dos concorrentes que se inscrevem como 'Individuais' e são de 'Valongo'.

c) imprimir o telemóvel e a prova em que está inscrito cada concorrente cujo nome seja 'Paulo' ou 'Ricardo', desde que seja da Vodafone.

d) imprimir os 20 primeiros registos com os nomes convertidos para minúsculas.

e) imprimir os 20 primeiros registos num novo ficheiro de output mas em formato Json.

2 Processador de Pessoas listadas nos Róis de Confessados

Construa agora um ou vários programas Python para processar o texto '`processos.txt`' com o intuito de calcular frequências de alguns elementos (a ideia é utilizar arrays associativos para o efeito) conforme solicitado a seguir

a) Calcula a frequência de processos por ano (primeiro elemento da data);

b) Calcula a frequência de nomes (considera um nome uma palavra e propaga o cálculo por todos os campos que contenham nomes);

c) Calcula a frequência dos vários tipos de relação: irmão, sobrinho, etc.

d) imprimir os 20 primeiros registos num novo ficheiro de output mas em formato Json.

3 Processador de Utilizadores registados no sistema Clav

Construa agora um ou vários programas Python para processar o texto '`clav-users.txt`' em que campos de informação têm a seguinte ordem: nome, email, entidade, nível, número de chamadas ao backend, com o intuito de calcular alguns resultados conforme solicitado a seguir:

- Produz uma listagem apenas com o nome e a entidade do utilizador, ordenada alfabeticamente por nome;
- Produz uma lista ordenada alfabeticamente das entidades referenciadas, indicando, para cada uma, quantos utilizadores estão registados;
- Qual a distribuição de utilizadores por níveis de acesso?
- Produz uma listagem dos utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome;
- Por fim, produz os seguintes indicadores:
  1. Quantos utilizadores?
  2. Quantas entidades?
  3. Qual a distribuição em número por entidade?
  4. Qual a distribuição em número por nível?

Para terminar, deve imprimir os 20 primeiros registos num novo ficheiro de output mas em formato Json.

4 BibTeXPro, Um processador de BibTeX

BibTeX é uma ferramenta de formatação de citações bibliográficas em documentos LaTeX, criada com o objectivo de facilitar a separação da base de dados da bibliografia consultada da sua apresentação no fim do documento LaTeX em edição. BibTeX foi criada por Oren Patashnik e Leslie Lamport em 1985, tendo cada entrada nessa base de dados textual o aspecto que se ilustra a seguir

```
@InProceedings{CPBFH07e,
  author =    {Daniela da Cruz and Maria João Varanda Pereira
               and Mário Béron and Rúben Fonseca and Pedro Rangel Henriques},
  title =     {Comparing Generators for Language-based Tools},
  booktitle = {Proceedings of the 1.st Conference on Compiler
               Related Technologies and Applications, CoRTA'07
               --- Universidade da Beira Interior, Portugal},
  year =      {2007},
  editor =    {},
  month =     {Jul},
  note  =     {}
}
```

De modo a familiarizar-se com o formato do BibTeX poderá consultar o ficheiro `exemplo-utf8.bib` que se anexa e ainda a página oficial do formato referido (`http://www.bibtex.org/`), devendo para já saber que a primeira palavra (logo a seguir ao caracter "@") designa a categoria da referência (havendo em BibTeX pelo menos 14 diferentes).

As tarefas que deverá executar neste trabalho prático são:

a) Analise o documento BibTeX referido acima e faça a contagem das categorias (`phDThesis`, `Misc`, `InProceeding`, etc.), que ocorrem no documento. No final, deverá produzir um documento em formato HTML com o nome das categorias encontradas e respectivas contagens.

b) Complete o processador de modo a filtrar, para cada entrada de cada categoria, a respectiva chave (a 1ªpalavra a seguir à chaveta), autores e título. O resultado final deverá ser incluído no documento HTML gerado na alínea anterior.

c) Crie um índice de autores, que mapeie cada autor nos respectivos registos, de modo a que posteriormente uma ferramenta de procura do Linux possa fazer a pesquisa.

d) Construa um Grafo que mostre, para um dado autor (definido à partida) todos os autores que publicam normalmente com o autor em causa.
Recorrendo à linguagem Dot do GraphViz[1], gere um ficheiro com esse grafo de modo a que possa, posteriormente, usar uma das ferramentas que processam Dot [2] para desenhar o dito grafo de associações de autores.

5 Conversor genérico de CSV para JSON

Trata-se de fazer um conversor de um qualquer ficheiro gravado em formato CSV (*Comma Separated Values*, original e tipicamente usado para descarregar uma Folha de Cálculo num ficheiro de texto) para o formato JSON[3](um formato textual neutro e muito simples, baseado no conceito de um conjunto de pares { "campo": "valor"}, concorrente do XML enquanto sistema de exportação/transferência de dados entre aplicações para assegurar a interoperabilidade).

Para poder realizar a conversão pretendida, é importante saber que a primeira linha do CSV dado funciona como cabeçalho que descodifica a que correspondem os valores que vem nas linhas seguintes. Até aqui nada de novo . . . , mas é claro que leva mais uns ingredientes.

O dataset dado poderá ter listas aninhadas nalgumas células. Mas nesse caso o cabeçalho terá um asterisco '*' a seguir ao nome do respetivo campo. Se nada mais for colocado o conversor deverá converter cada valor dessa coluna numa lista em JSON.

Mas a seguir ao aterisco pode haver uma função de agregação: `sum, avg, max, min`. Aí o conversor terá de aplicar a operação de fold respetiva sobre a lista e produzir o JSON de acordo.

[1] Disponível em `http://www.graphviz.org`
[2] Disponíveis em `http://www.graphviz.org/Resources.php` ou a ferramenta Web `http://www.webgraphviz.com/`
[3] Ver mais em `https://www.w3schools.com/js/js_json_intro.asp` ou `https://jsonformatter.curiousconcept.com/`