_____ / _____ / _____

Extrator de textos

Imagine que você tem um arquivo PDF contendo um texto formatado, cheio de desenhos e esquemas, além de "watermarks" (ou marca d'água). Seja como for, a disposição dos textos impede ou dificulta a captura dos textos através de um scanner e principalmente sua posterior conversão de imagem para texto.

Então, conhecendo como você conhece o formato interno de um arquivo PDF, você resolve construir um utilitário que:

- leia um arquivo PDF cujo nome é fornecido
- despreze toda a informação do arquivo que não se refira diretamente ao texto
- transcreva em um arquivo plano (editável pelo notepad) todo o texto que existir dentro do arquivo PDF
- informe ao final uma estatística de caracteres lidos e caracteres escritos no arquivo.

Preliminares Lembrando que a grande maioria dos arquivos PDF têm seus objetos de conteúdo devidamente comprimidos, você deve passar seu arquivo de entrada pelo utilitário PDFTK (ou similar), com vistas a descomprimir tudo. O comando é

```
PDFTK entrada.PDF output entdescomprimida.pdf uncompress
```

Neste segundo arquivo, uma boa coisa é que todo o conteúdo está distribuído em linhas terminadas por LF (hexadecimal `0A`). Daí, tais linhas podem ser lidas pelo comando C `fscanf` ou similar.

Processamento das linhas Uma vez obtida uma linha, por característica toda especial do formato PDF (que diz que os parâmetros de um comando vêm ANTES do comando), você deve começar a rastrear o comando de trás para a frente.

Um comando que deve ser desprezado, sem desprezar a linha toda é o comando `T*`. Quando ele for encontrado, a pesquisa prossegue como se `T*` não tivesse existido.

Em particular nos interessam apenas os comandos que mandam registrar textos e que são: `Tj`, `TJ` e `'`. Se os primeiros 1 ou 2 caracteres imediatamente antes do `0A` não forem estes 3 comandos, a linha deve ser desprezada. Se for um destes 3 comandos, o seu programa deve remover colchetes ([]'s) porventura existentes (no início e no final do comando), caso existam. Se for um `Tj` ou um `'` seu programa deve remover parênteses {()'s} existentes no início e no final da frase e depositar integralmente a frase restante no seu arquivo de saída.

processamento do kerning Se o comando encontrado for um `TJ`, a linha conterá, além dos caracteres a imprimir, informações de kerning.

Lembrando o kerning é o ajuste horizontal, para a esquerda a fim de ajustar duas letras que podem ou não ser complementares em termos de forma. O exemplo clássico é `A W`, que depois de receber o kerning fica `AW`.

Dentro do formato PDF os exemplos acima ficariam `(AW)` e `(A)505(W)`. O número que aparece fora dos parêntes é uma unidade de medida em milésimos de ponto (1 ponto = 0.341mm) que é subtraida da posição base da próxima letra. No exemplo acima, a letra `W` é aproximada 505 milésimos de ponto da letra `A`.

A regra para o seu programa processar esta uma linha com `TJ` é: Ao processar a linha da esquerda para a direita, sempre que aparecer um abre parênteses este deve ser desprezado e o próximo caracter e os seus subsequentes devem ser jogados na saída ATÉ encontrar-se um fecha parênteses, que também deve ser desprezado. Números que estejam fora dos parênteses significam ajuste de kerning e devem ser desprezados também.

☞ Para você fazer

Você deve escrever um programa que leia um arquivo PDF através da entrada padrão e que gere um arquivo plano na saída padrão contendo apenas os caracteres formadores do texto do arquivo PDF original, sem nenhma informação adicional de formatação.

A avaliação deste trabalho se fará através da entrega do programa fonte e do programa executável do compilador. Esta entrega deverá se fazer em mídia magnética no pendrive do professor. Os programas fonte e executável deverão ter o nome de □ Por favor, coloque como comentário logo no início do seu programa fonte o seu nome. O programa fonte será examinado quanto a ocorrência de contrafação. O programa executável será certificado quanto à correspondência com o programa fonte entregue e receberá uma carga de teste inédita, cujos valores esperados são conhecidos pelo avaliador.

□