

# Prova Final de Programação 1

## (IF968)

Fernando Castor  
Centro de Informática  
Universidade Federal de Pernambuco

16 de dezembro de 2015

**Instruções.** Durante a prova, é proibido usar telefone celular com qualquer fim, se comunicar com colegas, realizar consultas a qualquer tipo de material. A falha em observar essas restrições resultará em penalidades na nota da prova. Além disso, em caso de dúvida, indique na prova se identificou uma ambiguidade e como lidou com ela, para evitar de atrapalhar os outros alunos chamando o professor.

1. **(6,0 ptos.)** Crie uma função, `formatar()`, para formatar texto. Sua função recebe um string  $S$  e transforma esse string em uma lista de palavras. Cada palavra é um string, logo, o resultado da função `formatar()` será uma lista de strings. A função `formatar()` deve funcionar usando outras funções, que você deve implementar:

- `pegarPrimeiraPalavra()`: recebe um string de entrada e devolve um string contendo todos os caracteres encontrados nesse string até aparecer o primeiro espaço em branco. Por exemplo, `pegarPrimeiraPalavra("abc def")` devolve `"abc"`.
- `descartarPrimeiraPalavra()`: recebe um string de entrada e devolve um string contendo todos os caracteres encontrados nesse string após o primeiro espaço em branco, incluindo o próprio. Por exemplo, `descartarPrimeiraPalavra("abc def")` devolve `" def"`.
- `descartarBranco()`: recebe um string como entrada e remove desse string todos os espaços em branco que aparecem à esquerda desse string. Por exemplo, `primeiraPalavra(" ghi")` devolve `"ghi"`.

A função `formatar()` propriamente dita usa essas funções todas e funciona como no exemplo abaixo:

```
formatar("Finalmente vou ficar de férias de P1. Que alegria!")
```

```
["Finalmente", "vou", "ficar", "de", "férias", "de", "P1.", "Que", "alegia!"]
```

Não precisa se preocupar em remover os sinais de pontuação. Só os espaços em branco. Seu programa deve funcionar de maneira estritamente recursiva (nada de `for`, `while` ou compreensão de listas) e não pode usar funções como `split()` ou `replace()`.

2. **(5,0 ptos.)** Um arquivo representa um banco de dados de pessoas. Esse arquivo está estruturado da seguinte maneira:

```
123456789-00
Matt Murdock
Nova Iorque, EUA
```

```
666666666-69
Elektra Natchios
Atenas, Grécia
```

```
777888999-10
Wilson Fisk
Nova Iorque, EUA
```

```
...
```

Crie duas funções:

- `lerRegistros()`: lê todo o conteúdo do arquivo, guardando os elementos em um **dicionário** na memória, usando o CPF como chave para cada registro. Cada um desses registros deve agrupar todas as informações sobre cada pessoa (nome, endereço e país). Seu programa deve ignorar as linhas em branco entre os registros.
- `atualizar()`: recebe um CPF, um nome, um endereço e um país e atualiza os dados de um registro pré-existente (para o CPF fornecido) fornecidos como argumento. Essa atualização deve se refletir no arquivo. O arquivo final deve conter apenas **um** registro relativo a esse CPF, ligado às informações passadas para a função `atualizar()`, além de todos os registros pré-existent, associados a CPFs diferentes.