

Prova Final de Programação 1 (IF968)

Fernando Castor
Centro de Informática
Universidade Federal de Pernambuco

17 de julho de 2017

INSTRUÇÕES. A prova deverá ser realizada individualmente. Qualquer forma de comunicação entre estudantes será punida com nota 0 (ZERO). A prova deverá ser realizada nos computadores do laboratório. Assim que receber este documento, você deve desligar o cabo de rede e manter abertos apenas este documento e o IDLE. Sua solução deve estar em um arquivo cujo nome é *seulogin.py*. Por exemplo, se o professor fosse resolver a prova, a colocaria em um arquivo chamado `fjclf.py`. Ao terminar a prova, levante a mão para que o professor veja que você acabou. A partir desse momento, é **proibido** modificar suas respostas. Reconecte o cabo de rede e envie suas respostas para os endereços de email abaixo:

- castor@cin.ufpe.br

O assunto da sua mensagem deve ser [IF968]. Em todas as questões dessa prova, é proibido usar as funções tipicamente proibidas: `split`, `remove`, o operador `in`, etc. Na dúvida, pergunte ao professor. Como sempre, a diretriz geral é: se fica fácil demais usando uma função, provavelmente não é permitido usá-la.

As questões estão na próxima página.

1. **(2,5 ptos.)** Um palíndromo é uma palavra que é lida da mesma forma tanto no sentido tradicional de leitura quanto de trás para frente. Exemplos palíndromos incluem “arara”, “ovo”, “radar” e “omissíssimo”. Construa uma **função recursiva chamada palindromo** que, dada uma palavra (sem acentos), determina se ela é um palíndromo. Por exemplo:

```
>>> palindromo("arara")
True
>>> palindromo("omississimo")
True
>>> palindromo("ararinha")
False
```

2. **(2,5 ptos.)** Relembrando, da primeira aula do curso, o algoritmo de Heron da Alexandria para calcular a raiz quadrada de um número:

1. Comece com um palpite qualquer, r
2. Se $r * r$ é **próximo o suficiente**, pare e diga que r é a resposta
3. Caso contrário, faça um novo palpite usando a média entre r e N/r
4. Chame esse novo palpite de r e volte ao passo 1

Implemente uma **função recursiva raizQuadrada** que, dados um número e uma margem de erro, calcula a raiz quadrada desse número usando o algoritmo apresentado acima. É proibido usar laços (**for**, **while**) para resolver essa questão.

3. **(2,5 ptos.)** Implemente uma função chamada **converterParaDecimal** que, dado um string correspondendo a um número em base 2 (ou seja, contendo apenas 0s e 1s), converte esse número para a base 10. Converter um número da base 2 para a base 10 é fácil. Por exemplo, dado o número 1101, para convertê-lo para base 10, realizamos a soma $1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$ que é igual a 13. De forma mais geral, se o número em base 2 tem N dígitos, teremos uma soma $2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0$ sendo que cada termo dessa soma é multiplicado pelo elemento que aparece na mesma posição no número em base 2 e esse número pode ser apenas 0 ou 1. Mais exemplos a seguir:

```
>>> converterParaDecimal('10011')
19
>>> converterParaDecimal('1111010111')
983
>>> converterParaDecimal('1010011010')
666
```

No primeiro exemplo acima, 1 é o índice do elemento 4. Entre os elementos da lista, o elemento maior que 4 que está mais próximo é o número 5. No segundo exemplo, 2 é o índice do elemento 3. O elemento mais próximo a 3 que é maior que ele é o número 6, que aparece na posição 1. Note que o maior mais próximo pode aparecer antes ou depois da posição i . No terceiro exemplo, a função emite uma mensagem de erro porque não há elementos maiores que 10 na lista. O quarto exemplo mostra uma situação de empate, pois dois elementos maiores que 10 (o elemento da posição 1) estão à mesma distância dele. Neste caso, a função pode devolver qualquer um deles. É proibido usar laços (**for**, **while**) para resolver essa questão.

4. **(2,5 ptos.)** Implemente uma função chamada **subPalindromo** que, dado um string, devolve o seu maior sub-string que é um palíndromo. Se o tamanho do maior sub-string que é um palíndromo for 1, sua função pode devolver qualquer caractere do string. Exemplos:

```
>>> subPalindromo("Fernando")
'nan'
>>> subPalindromo("omississimo")
'omississimo'
>>> subPalindromo("palindromo")
'omo'
>>> subPalindromo("desse")
'esse'
>>> subPalindromo("genus")
'g'
```

Dica: lembre-se que todo palíndromo começa e termina com a mesma letra!