

# Prova 2

## Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

30 de outubro de 2014

Nome: \_\_\_\_\_

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca **prova2.h**), função **main**, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- As funções implementadas no módulo **prova2.h** podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo **prova2.h**, considere que ela está implementada de forma correta.

1. (26 points) Para as questões a seguir, considere que as implementações serão feitas no módulo "prova1.h".

a. (3 pts) Escreva uma função que recebe uma *string* como parâmetro e retorna o seu tamanho. A sua função deve ter o seguinte protótipo:

```
int tamStr(char s[]);
```

b. (6 pts) Escreva uma função que recebe uma *string* *s* como parâmetro e retorna 1 se *s* for um palíndromo e 0 caso contrário. Um palíndromo é uma palavra, frase ou qualquer outra sequência de caracteres que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. As palavras *radar*, *osso* e *reviver* são exemplos de palíndromos. A sua função deve ter o seguinte protótipo:

```
int ehPalindromo(char s[]) ;
```

c. (5 pts) Escreva uma função que recebe uma cadeia de caracteres *s* como parâmetro e a preenche com uma palavra lida do teclado. Considere que o usuário terminou de digitar a palavra quando ele apertar a tecla **ENTER** ou **ESPAÇO**. A palavra deve ser armazenada no formato de uma *string*. A sua função deve ter o seguinte protótipo:

```
void lePalavra(char s[]) ;
```

d. (2 pts) Defina um novo tipo de dados para representar uma palavra. Esse tipo de dados deve ser chamado de **Palavra** e deve ter campos para (i) armazenar o texto da palavra, (ii) indicar se a palavra é um palíndromo ou não e (iii) armazenar o tamanho da palavra.

e. (4 pts) Escreva uma função de nome **preenchePalavra** que recebe um parâmetro do tipo **Palavra** como parâmetro **por referência** e preenche os seus campos. Para isso, você deve utilizar as funções **lePalavra**, **ehPalindromo** e **tamStr**.

f. (6 pts) Escreva uma função que recebe como parâmetros um vetor de **Palavras** e a quantidade *n* de **Palavras** nesse vetor e retorna o tamanho médio das palavras armazenadas no vetor.

## 2. (6 points)

Escreva um programa para ler 10 palavras do usuário e imprimir todos os palíndromos de tamanho maior que a média dos tamanhos das palavras lidas. Exemplo: se o usuário entrar com as palavras *ba*, *bb*, *bi*, *bo*, *bu*, *ca*, *cc*, *ci*, *co* e *bob*, o seu programa deve imprimir somente *bob*, pois *cc* e *bb* são palíndromos de tamanho menor que a média, que é 2.1.