

P3 Programação II 2013.2 Departamento de Informática /PUC-Rio

QUESTÃO 1

Considere a representação de polinômios $a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$ através de uma lista encadeada, onde cada elemento tem um inteiro para a potência do termo, um float para o coeficiente e uma referência para o próximo termo do polinômio. Considere também que a lista encadeada é representada por um ponteiro para o seu primeiro elemento.

(1a) Defina o tipo estruturado que representa o elemento da lista.

(1b) Usando o tipo definido na questão anterior, escreva a função `lista_insere` (sem `#include`) que insere um novo termo do polinômio, retornando a lista atualizada ou NULL se ocorrerem problemas. Os elementos na lista não precisam estar ordenados por expoente.

(1c) Usando a função definida na questão anterior, escreva a função `derivada` que recebe um polinômio e retorna um novo polinômio que é a derivada do polinômio original. A derivada de um polinômio é dada pela soma das derivadas de cada um de seus termos e a derivada de um termo $a_n x^n$ é dada por $n a_n x^{n-1}$. Por exemplo, a derivada do polinômio $3x^2 - 2x + 1$, que é representado por uma lista com 3 termos, é o polinômio $6x - 2$.

QUESTÃO 2

Uma cadeia de caracteres é um palíndromo se ela é lida da mesma forma da esquerda para a direita e da direita para a esquerda, por exemplo: “rever”, “radar” e “72127”. Escreva uma função chamada `palindromo` que testa se uma cadeia de caracteres é um palíndromo. Mas ao invés de usar funções similares às encontradas na biblioteca de *strings* da linguagem C, implemente uma solução que use uma pilha de caracteres. Gerencie limpeza de memória e interrompa a verificação tão logo encontre uma situação que invalide a condição de ser um palíndromo. Considere o seguinte TAD em sua solução:

```
typedef struct pilhac Pilhac;
/* constroi e destrói */
Pilhac * pilhac_cria(void);
void pilhac_libera(Pilhac * p);
/* operadores */
void pilhac_push(Pilhac * p, char a);
char pilhac_pop(Pilhac * p);
int pilhac_vazia(Pilhac * p);
```

Você não precisa implementar as funções do TAD acima. Você também não precisa considerar situações que exigem mensagens de erro e/ou o uso da função `exit()`.

QUESTÃO 3

Considere uma árvore binária de busca que guarda palavras em ordem crescente sem repetições:

```
struct noArv {
char nome[50]; struct noArv * esq; struct noArv * dir;
}; typedef struct noArv NoArv;
```

Escreva uma função `mostraPalindromo` que recebe uma árvore deste tipo e uma cadeia de caracteres `nome` e imprime, em ordem crescente, todas as palavras nesta árvore que antecedem alfabeticamente a cadeia `nome` e que são palíndromos. Sua solução deve levar em conta a ordenação da árvore. Você deve, obrigatoriamente, usar a função `palindromo` da questão anterior, mesmo que você não a tenha escrito. Você pode usar funções da biblioteca

string da linguagem C. Como exemplo, para `nome` igual a "radar" e a árvore abaixo, esta função imprime *ama arara osso radar*.

