O TAD Lista

Este trabalho visa implementar um Tipo Abstrato de dado (TAD) “lista de inteiros” usando listas encadeadas.

Atividade

Você deve implementar um TAD “lista de números inteiros” usando lista encadeada dupla não-circular.

As seguintes operações devem ser suportadas:

- `cria (lista)`: cria uma nova lista
- `destrói (lista)`: esvazia e destrói a lista, liberando a memória
- `insere (lista, item, posição)`: insere um novo item na posição indicada
- `item = retira (lista, posição)`: retira o item da posição indicada
- `item = consulta (lista, posição)`: informa o valor do item na posição indicada, sem retirá-lo
- `posição = procura (lista, valor)`: informa a posição da 1ª ocorrência do valor informado na lista
- `tamanho (lista)` : informa o número de itens na lista
- `imprime (lista)` : imprime o conteúdo da lista

A posição de inserção/remoção/consulta é um número inteiro positivo, onde 0 indica a primeira posição da lista.

Os arquivos necessários para desenvolver este trabalho estão

neste arquivo

. Eles são:

- `lista.h`: arquivo de cabeçalho com os protótipos das funções (**não deve ser alterado**).
- `lista.c`: arquivo que implementa as operações sobre as listas (“esqueleto” a completar).
- `tp4.c`: código que testa a biblioteca (**não deve ser alterado**).
- `tp4.txt`: saída esperada da execução de `tp4`
- `makefile`: arquivo do utilitário “make” para compilar seu código.



Você pode testar se a saída de seu programa está correta usando o comando `diff`:

```
$ ./tp4 > saida.txt
$ diff tp4.txt saida.txt
```

Esse comando mostra as diferenças entre dois arquivos de texto. Se ele não mostrar nada, os arquivos são iguais.

Entregáveis

Entregue um único arquivo `tp4.tgz` que contenha por sua vez os seguintes arquivos:

- `lista.h`: o mesmo arquivo fornecido, **não o modifique**
- `lista.c`: sua implementação das funções definidas em `lista.h`
- `tp4.c`: o mesmo arquivo fornecido, **não o modifique**
- `makefile`

Critérios de avaliação:

- Funcionar corretamente 💪
- Não ter problemas de memória (testar com Valgrind)
- Respeitar o conceito de TAD
- Usar funções para evitar repetição de blocos de código
- Respeitar estilo (endentação, espaçamento, nomes de variáveis)
- Usar comentários (nem demais, nem de menos)
- ... (outros que se fizerem necessários)

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=prog1:tad_lista

Last update: **2024/10/23 10:55**