

CI1001 - Programação I

André Grégio, Fabiano Silva, Luiz Albini e Marcos Castilho
Departamento de Informática – UFPR, Curitiba/PR

Quinta lista de exercícios

O objetivo deste trabalho é praticar programação em linguagem C. Para isso usaremos conceitos já conhecidos da disciplina de Algoritmos e Estruturas de Dados I, para podermos focar somente nos aspectos de implementação nesta nova linguagem.

O trabalho consiste em criar uma biblioteca de funções em C que manipulam o Tipo Abstrato de Dados *tad_pilha*. Os protótipos das funções que queremos são as seguintes (considerando uma pilha de números inteiros):

Observação: os protótipos foram propositalmente escritos usando-se a passagem de parâmetros por cópia. Vocês devem adaptá-los para que as funções modifiquem a estrutura da pilha, isto é, faz parte do exercício aprender a passar endereços nos parâmetros!

```
// faz a inicialização do TAD pilha
void inicializa_pilha (tad_pilha p) {

// retorna 1 se a pilha p esta vazia e 0 caso contrario
int pilha_vazia (tad_pilha p) {

// retorna o numero de elementos da pilha p
int tamanho_pilha (tad_pilha p) {

// empilha o inteiro x na pilha p
// retorna 1 se a operacao foi realizada com sucesso e 0 caso contrario
int empilha (int x, tad_pilha p) {

// retorna em t o elemento do topo da pilha e o desempilha
// a funcao retorna 1 se a operacao foi bem sucedida e 0 caso contrario
int desempilha (int t, tad_pilha p) {

// retorna em t o elemento do topo da pilha, sem desempilhar
// a funcao retorna 1 se a operacao foi bem sucedida e 0 caso contrario
int topo (int t, tad_pilha p) {
```

Para este exercício, considerar que a implementação do *tad_pilha* utiliza um registro (em *Pascal* é um *record* e na linguagem *C* é uma *struct*) que contém dois campos:

- o início da pilha (topo)
- um vetor de inteiros que contém os elementos da pilha

Basicamente, é o mesmo exercício feito no semestre passado, porém com código em C e com um comportamento das funções que permite o tratamento de erros. Por exemplo, a nova função *desempilha* devolve o elemento a ser desempilhado como um parâmetro e seu retorno é um inteiro que contém um código informando se a operação teve sucesso ou não. Por exemplo, essa função retorna um código de erro (0) quando se tenta desempilhar uma pilha que já está vazia.

Os detalhes do trabalho seguem abaixo:

1. Para definir o *tad_pilha* corretamente em C, vocês devem ler o excelente material produzido pelo professor Maziero e entender como usar a instrução `typedef` em C, que é equivalente ao `type` do Pascal:
<http://wiki.inf.ufpr.br/maziero/doku.php?id=prog2:estruturas>;
2. Neste texto vocês vão aprender a definir uma estrutura que é baseada no conceito de `struct`;
3. Baixe o arquivo *tad_pilha.c*. É nele que você vai trabalhar e é esse arquivo que deverá ser submetido ao final da aula;
4. Observe que as chamadas para as funções também devem ser modificadas para atender a questão da passagem de endereços como parâmetros!
5. Este código pode ser encontrado em:
http://www.inf.ufpr.br/cursos/ci1001/Aulas/tad_pilha.c;
6. O seu programa deve ser compilado com as opções `-ansi` e `-Wall`;
7. Seu programa deverá ser testado antes do envio pela comparação da sua saída com a saída esperada (usando o comando *diff* e pipes), que pode ser obtida em:
http://www.inf.ufpr.br/cursos/ci1001/Aulas/saida_tad_pilha.txt;