

Estrutura de Dados

Exercícios: *alocação dinâmica de memória*

1 Ambiente de desenvolvimento

1. Baixe, instale e configure um compilador C, juntamente com o editor ou IDE de sua preferência, de forma a já ficar desde o início com um ambiente de desenvolvimento devidamente pronto para o restante do curso.
2. Veja na sala do Moodle os *links* dos programas para Windows e Linux.

2 Instruções

1. Desenvolva os exercícios abaixo em C, utilizando como referência as informações apresentadas na aula e a apostila de C que se encontra no Moodle.
2. O objetivo desta lista de exercício é desenvolver um novo tipo *string* utilizando *structs*, ponteiros e alocação dinâmica de memória.
3. Crie um módulo `my_str.h` e `my_str.c` com guarda de inclusão.
4. Crie um programa cliente para testar a sua nova biblioteca de *string*;

3 Exercícios

1. Utilizando `struct` crie um novo tipo de dado que permite armazenar um vetor de `char` de tamanho variável e o seu tamanho atual.
2. Se achar útil, utilize o comando `typedef` para criar um *alias* para a estrutura criada no item anterior.
3. Faça uma função que cria (aloca dinamicamente) uma nova *string* vazia.
4. Faça uma função que preenche uma nova *string* a partir de um vetor de `char` e o seu tamanho. Exemplo de um possível cabeçalho:

```
void fill(String *s, char *c, int len);
```

Obs.: Você deve fazer uma cópia do vetor original. Para isso, use as funções da biblioteca `string.h`.

5. Faça uma função que exibe a sua nova estrutura de *string* na tela.
6. Faça uma função para destruir (desalocar da memória) um ponteiro para seu tipo *string*.
7. Faça uma função para concatenar duas *strings*. As *strings* originais não devem ser modificadas.