
strlib: string library

Neste VPL você deverá desenvolver uma biblioteca *simples* de manipulação de *strings*. Sim, já existe uma biblioteca de manipulação de *strings* padrão em C, a `string.h`. Sua biblioteca não será tão completa nem tão complexa como a `string.h`. Este VPL possui dois objetivos principais: (i) aumentar sua compreensão acerca de *strings* e como elas funcionam em C e (ii) praticar ponteiros e alocação dinâmica.

Alguns pontos importantes que você deve ter em mente durante a realização deste VPL:

- Você *não* pode utilizar nenhuma função pronta da biblioteca `string.h`. De fato, você *não* deve nem incluir esta biblioteca neste VPL. Caso isto ocorra, você receberá nota zero.
- Um `char*` é um ponteiro para `char`, ou seja, ele é um ponteiro que pode apontar para regiões do tipo `char` (com um ou mais caracteres). Neste sentido, um `char*` pode ser considerado uma *string*, visto que aponta para o primeiro caractere da *string* (ele armazena, de fato, este endereço).
- Toda e qualquer *string* deve, *obrigatoriamente*, ser terminada com o caractere `\0`. Dessa forma, se você quiser armazenar a palavra "Galaxie" em uma *string*, precisa de um vetor de `char` com 8 posições.

0	1	2	3	4	5	6	7
G	a	l	a	x	i	e	\0

Neste caso, o comprimento da *string* é 7, mas internamente, precisamos de 8 posições para armazená-la (ou seja, você precisa de um vetor de 8 posições). Em resumo, o tamanho da *string* não leva em conta o `\0`, mas ele precisa estar armazenado!

- Qualquer *string* literal¹ tem, automaticamente, o caractere `\0` adicionado ao seu final. Por exemplo, ao chamar a função `str_length` (veja a descrição desta função no VPL) como:

```
1 int tamanho = str_length("Abacaxi");
```

o compilador irá adicionar o caractere `\0` ao final de `Abacaxi`. Assim, quando a chamada da função for de fato realizada, o caractere `\0` estará lá.

- Use e consulte a tabela ASCII para realizar esse VPL, ela será muito útil!
- Por fim, lembre-se que um único caractere e uma *string* são tratados diferentemente em C. Um caractere único é denotado por aspas simples (ex, `'a'`) enquanto uma *string* é denotada por aspas duplas (ex, `"a"`). Neste caso, `'a'` é diferente de `"a"`.

Essas são as principais dicas para realizar este VPL. Dúvidas? Procure um dos livros da ementa (veja o capítulo de ponteiros do C Completo e Total), o monitor e o professor da disciplina. Boa programação!

¹Uma *string* literal ou *string* constante é aquela usada entre aspas duplas. Por exemplo, no comando `printf("Ola Mundo")`, `Ola Mundo` é uma *string* literal.