

Algoritmos e Programação de Computadores
Projeto 05: Funções
ABI/LFI/TAI

Prof. Daniel Saad Nogueira Nunes



1 Introdução

Funções são estruturas que executam um bloco de código quando invocadas. Possuem papel fundamental na modularização de programas em C, o que possibilita a construção de programas maiores, mais fáceis de serem lidos e com maior manutenibilidade.

Neste projeto uma série de funções deverão ser implementadas para exercitar o conceito.

2 Especificação

Neste projeto deverão ser implementadas as seguintes funções em um arquivo `.c`:

- `void getline(char str[], int n);`: faz a leitura de uma linha da entrada e coloca o resultado em `str`. A leitura deve parar quando: o limite de $n - 1$ caracteres for lido (o `'\0'` deve constar no final da string), ou o caractere de quebra de linha for encontrado. A próxima chamada a `getline` deve continuar a leitura de onde a chamada anterior parou.
- `int menor(char str1[], char str2[], int n1, int n2);`: recebe duas strings, `str1` e `str2`, de tamanhos `n1` e `n2`, e retorna verdadeiro, se `str1` é menor que `str2` e falso, caso contrário. O alfabeto a ser considerado para fins de ordenação é o da tabela ASCII. O tamanho de uma string é definido como o número de caracteres até o primeiro `'\0'`.
- `void tokenize(char str[], int n, char delimitador);`: recebe uma string `str` de tamanho `n` e um caractere `delimitador` e imprime todas as strings delimitadas por este caractere em **uma linha cada**. Por exemplo, se a string for `"abracadabra"` e o caractere delimitador é `'a'`: deverão ser impressas as strings: `br,c,d,br`. O tamanho de uma string é definido como o número de caracteres até o primeiro `'\0'`.
- `void retard(char str[], int n);`: recebe uma string `str` de tamanho `n` e converte todas as suas vogais para minúsculo e suas consoantes para maiúsculo. Por exemplo, se a string de entrada for `"87 eh do Sport"`, a string deve ser modificada para `"87 eH Do SPoRT"`. O tamanho de uma string é definido como o número de caracteres até o primeiro `'\0'`.

Restrição: as funções dos cabeçalhos `string.h` e `ctype.h` não deverão ser utilizadas, com exceção da função `strlen`, cuja utilização é opcional. Além disso, as funções não deverão alterar as entradas (exceto se disposto em sua descrição).

O arquivo `.c` não deverá possuir a presença da função `main`, apenas os **protótipos** e as **implementações** das funções.

2.1 Testes

Na correção dos trabalhos, as funções serão inseridas em um arquivo `.c` com função `main` que faz uma série de invocações das funções implementadas, através de vários conjuntos de dados, para averiguar que estão corretamente implementadas.

Na fase de desenvolvimento, é recomendável o teste das funções da mesma forma, mas ressalta-se que o arquivo `.c` a ser entregue não deve possuir nada além dos protótipos e a implementação das funções.

2.2 Documentação

O código deverá ser devidamente indentado, documentado através de comentários e respeitando as boas práticas de programação considerando os nomes das variáveis, caso contrário, haverá desconto na nota do projeto.

Além disso, no cabeçalho do arquivo fonte deverá constar o nome e matrícula do aluno.

3 Considerações

- Este projeto deve ser executado **individualmente**.
- A incidência de plágio acarretará automaticamente nota 0 (**zero**) para os envolvidos. Medidas disciplinares também serão tomadas.
- Trabalhos que não compilarem não serão avaliados.
- O código deve ser entregue em uma pasta zipada com a devida identificação do aluno através da sala de aula virtual da disciplina.