

Algoritmos e Programação I: Lista 07

Marcelo Hashimoto

19 de abril de 2013

Todo programa pedido nesta lista deve apresentar a seguinte estrutura:

```
#include <stdio.h>

#define TAM 100

int main() {
    char s[TAM];

    /* variáveis e vetores adicionais, se necessário */

    printf("Digite uma linha de texto: ");
    fgets(s, TAM, stdin);

    /* código específico do exercício */

    return 0;
}
```

Ou seja, começa pedindo para o usuário digitar uma linha de texto (sequência de caracteres que termina em uma quebra de linha) e usa a função `fgets` para armazenar essa linha de texto como *string* no vetor de caracteres `s`.

- **Observação 1.** Note que nesse vetor cabem no máximo `TAM` caracteres. Portanto, cabe no máximo uma *string* de comprimento `TAM - 1`, visto que a presença do caractere `'\0'` é obrigatória.
- **Observação 2.** O segundo parâmetro da função `fgets` garante que, se o usuário digitar uma linha de texto muito longa, somente os `TAM - 1` primeiros caracteres dela serão considerados. Portanto, não há risco de estouro.
- **Observação 3.** A função também considera a quebra de linha como parte da linha de texto digitada. Por simplicidade, o restante do programa pode fazer o mesmo: considere o `'\n'` como um caractere qualquer.
- **Observação 4.** Não se preocupe com o significado de `stdin` por enquanto.

1. Escreva um programa que pede para o usuário digitar uma linha de texto e imprime essa linha.
Escreva duas versões da impressão: uma que execute um único `printf` com `%s` e uma que execute uma repetição de `printfs` com `%c`.
2. Escreva um programa que pede para o usuário digitar uma linha de texto e imprime o *comprimento* dessa linha.
3. Escreva um programa que pede para o usuário digitar uma linha de texto e imprime essa linha *invertida*, ou seja, imprime o último caractere, depois o penúltimo, o antepenúltimo e assim em diante.