

## Atividade de Programação 2 - Concatenação de Strings

Algoritmos e Estruturas de Dados I

Tempo de execução limite: 1s

A concatenação de *strings* é um processo para anexar uma sequência de caracteres no final de outra sequência de caracteres.

A contagem de caracteres consiste em saber quantas vezes um determinado caractere é encontrado na *string*.

Deve-se escrever um programa que aborde essas duas características.

### Entrada:

A entrada de dados para cada teste inicia-se com uma linha contendo três valores: o tamanho máximo de caracteres da primeira *string*  $X$  ( $1 \leq X \leq 80$ ), o tamanho máximo de caracteres da segunda *string*  $Y$  ( $1 \leq Y \leq 80$ ), e um caractere  $C$  do tipo *char*.

Na próxima linha, devem ser fornecidos os dois *strings*, separados por um espaço em branco. Nota-se que o tamanho máximo ( $X$  ou  $Y$ ), não representam necessariamente o tamanho real da *string* digitada, a qual pode ser menor que a sua capacidade máxima.

### Saída:

A saída do programa deve ser apresentada em três linhas. A primeira contém uma *string*  $Z$  que corresponde às strings  $X$  e  $Y$  invertidas e concatenadas.

A segunda linha mostra o tamanho real da *string*  $Z$ .

Na terceira linha é exibida a contagem de quantas vezes aparece o caractere  $C$ , na *string*  $Z$ . Nota-se que o tamanho real de  $Z$  é a soma do tamanho real das duas *strings* e não necessariamente o tamanho alocado ( $X+Y$ ).

### Dicas:

- Deve-se lembrar que toda *string* em C/C++ contém o caractere especial `'\0'` para demarcar o término da *string*. Deve-se levar este fato em consideração na alocação do vetor de caracteres, e nas manipulações para determinação de tamanhos e cópias.

### Restrições:

- O programa deve ser escrito em C ou C++;
- Deve-se utilizar ponteiros no programa, para representar a *string*  $X$  e a *string*  $Y$ , as quais devem ser alocadas de forma dinâmica com a função `malloc(...)`;
- Caso o código-fonte seja codificado em C, deve-se utilizar as bibliotecas padrões, com exceção de algumas funções de `<string.h>`, que não devem ser usadas: `strlen`, `strcat`, `strncat`, `strpbrk`;
- Caso seja em C++, deve-se utilizar as bibliotecas padrões, com exceção de `<string>` e `<sstream>`, e algumas funções de `<cstring>`, como: `strlen`, `strcat`, `strncat`, `strpbrk`;

Exemplo de Entrada	Exemplo de Saída
8 7 a programa teste01	amargorp10etset 15 2

Exemplo de Entrada	Exemplo de Saída
15 7 n Caderno azul	onredaCluza 11 1

Exemplo de Entrada	Exemplo de Saída
10 7 9 folha verde	ahlofedrev 10 0