

Laboratório 10

Cadeia de caracteres (strings).

Introdução à Programação C (CMT012)

Prof. Ronald Souza

IC/UFRJ — 25/10/2023

Objetivo

Praticar o uso de operações sobre cadeia de caracteres (strings).

Todos os seus programas devem preencher a estrutura abaixo.

```
/*
Autor: <nome do aluno>
Data: <data de hoje>
Descrição: <o que o programa faz>
Entrada: <o que o programa espera receber como entrada>
Saída: <o que o programa retorna para o usuário>
Defesa: <restrições sobre os dados de entrada, se existirem>
*/

#include <stdio.h>

int main() {
    //dicionário de dados

    //corpo do programa

    return 0;
}
```

Atividade 1 (baseada no problema “Vestibular” da OBI-Junior 2008):

Parte das universidades brasileiras ainda usa o vestibular para selecionar seus alunos. O vestibular consiste de uma ou mais provas sobre as matérias do Ensino Médio, visando avaliar os conhecimentos dos candidatos. Um formato popular de prova de vestibular é a prova objetiva. Neste formato, cada candidato deve escolher uma das cinco alternativas apresentadas pela questão como sendo a correta. Durante a correção dos cartões, cada questão onde a alternativa escolhida pelo candidato é a mesma do gabarito, ele ganha um ponto. Escreva um programa que, dado o gabarito e as respostas de um dos candidatos, determina o número de acertos daquele candidato.

Entrada: A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão. A primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 80$), representando o número de questões na prova. A segunda linha da entrada contém uma cadeia de N caracteres, indicando o gabarito da prova. A terceira linha da entrada contém outra cadeia de N caracteres, indicando as opções marcadas pelo candidato. Ambas as cadeias contém apenas os caracteres ‘A’, ‘B’, ‘C’, ‘D’ e ‘E’ (sempre em letra maiúscula).

Saída: Seu programa deve imprimir na saída padrão uma única linha contendo um único inteiro, indicando o número de acertos do candidato.

Obs.: Para ler as cadeias de caracteres use o formato %s.

Teste 1

Entrada	Saída
7 AEDBCCE ADDCCBE	4

Teste 2

Entrada	Saída
10 ABCDEABCDE BCDEABCDEA	0

Atividade 2: Os alunos de uma turma fizeram duas provas. Escreva um programa que leia do teclado **uma única string** contendo o **nome completo** e **as duas notas** de um aluno, separados por espaço, calcule a média aritmética dessas duas notas e imprima em seguida o nome do aluno e a média obtida. Use a função *'sscanf'*, vista em nossa última aula.

Permita que o programa aceite novas entradas até que uma entrada inválida seja fornecida.

Atividade 3: Escreva um programa em C que leia do teclado **uma única string** no formato

<código> <preço> <desconto>

com: (1) código de um produto (palavra que pode conter letras e números); (2) preço do produto (valor em reais); e (3) percentual de desconto do produto. Note que os 3 campos são separados por um espaço. O programa deverá calcular o novo valor do produto com o desconto dado, gerar uma nova string no formato

<código> <preço com desconto>

e exibir essa string na tela.

Use as funções *sscanf* e *sprintf*.

Atividade 4: Implemente a função *copiaString* definida abaixo. Essa função recebe duas strings como entrada e copia a primeira string para a segunda, até o tamanho máximo do espaço de memória reservado para a segunda string.

```
void copiaString(char str1[], char str2[], int max);
```

Execute o programa abaixo para avaliar a sua implementação:

```
int main() {
    char palavra1[20], palavra2[10];
    puts("Digite uma palavra com ate 19 caracteres:");
    scanf("%19[^\n]", palavra1);
    copiaString(palavra1, palavra2, 9);

    printf("%s\n%s\n", palavra1, palavra2);
    return 0;
}
```

Exemplo de entrada e saída

Entrada	Saída
ola mundo ola	ola mundo ola ola mund

→ Agora inverta o tamanho das palavras: *palavra1* com tamanho 10 e *palavra2* com tamanho 20, e **verifique a corretude da sua implementação.**

Exemplo de entrada e saída

Entrada	Saída
ola mundo	ola mundo ola mundo