

Aula 25: Strings: vetor de char terminado em \0

Lógica de Programação – Superior · Bloco 3

José Dihego · IFBA

O que você precisa saber hoje

Objetivo: Ler, percorrer e comparar caracteres de uma string sem bibliotecas extras.

- `char s[100];` guarda até 99 letras + o `'\0'` final
- Uma palavra: `scanf("%99s", s);` uma linha: `fgets(s, sizeof s, stdin)`
- Percorrer: `for (i = 0; s[i] != '\0'; i++)`
- Caracteres são números (ASCII): letras são consecutivas, `'a' < 'b'`
- Minúscula de uma maiúscula: `c - 'A' + 'a'` (escreva sua função)

```
int minuscula(char c) {  
    if (c >= 'A' && c <= 'Z') return c - 'A' + 'a';  
    return c;  
}  
  
/* no main */  
int ordenada = 1;  
for (int i = 1; s[i] != '\0'; i++)  
    if (minuscula(s[i - 1]) > minuscula(s[i]))  
        ordenada = 0;
```

Erros que custam pontos

Errado	Certo	Por quê
<pre>if (s == "sim") scanf("%s", &s); char s[5] = "maçãs"; if (c >= 'A' c <= 'Z')</pre>	<pre>/* compare letra a letra */ scanf("%99s", s); char s[20]; if (c >= 'A' && c <= 'Z')</pre>	== compara endereços, não textos o vetor já é um endereço faltou espaço para o \0 (e acentos ocupam 2 bytes) com todo caractere passa

```
char s[] = "FLoRa"; int ordenada = 1;
for (int i = 1; s[i] != '\0'; i++) {
    char a = s[i - 1], b = s[i];
    if (a >= 'A' && a <= 'Z') a = a - 'A' + 'a';
    if (b >= 'A' && b <= 'Z') b = b - 'A' + 'a';
    if (a > b) ordenada = 0;
}
```

i	s[i-1]	s[i]	ordenada
1	F	L	1
2	L	o	1
3	o	R	1
4	R	a	0

Avaliação 3 – 2023.2, Questão II

avaliacao_3_ano_2023_2.pdf

Palavra ordenada (letras em ordem alfabética, sem diferenciar maiúsculas)

Estratégia:

1. Ler a palavra com `scanf("%99s", p)`
2. Comparar cada letra com a anterior, ambas em minúscula
3. Uma inversão basta para "não ordenada"

Enunciado completo (com figuras e exemplos) no PDF da prova.

Solução (1/2)

```
1  #include <stdio.h>                                // printf e scanf
2
3  int minuscula(char c) {                             // 'A'..'Z' vira 'a'..'z'; o resto fica igual
4      if (c >= 'A' && c <= 'Z') return c - 'A' + 'a'; // na tabela ASCII, as letras são
        consecutivas
5      return c;                                       // já é minúscula (ou não é letra)
6  }
```

Solução (2/2)

```
8  int main(void) {                                // início do programa
9      char p[100];                                // a palavra (até 99 letras + '\0')
10     printf("Palavra: ");                          // pede a palavra
11     scanf("%99s", p);                             // %99s: lê uma palavra sem estourar o vetor
12     int ordenada = 1;                             // supõe ordenada até provar o contrário
13     for (int i = 1; p[i] != '\0'; i++) {          // compara cada letra com a anterior
14         if (minuscula(p[i - 1]) > minuscula(p[i])) ordenada = 0; // 'r' > 'o' em "Flro": fora
15         // de ordem
16     }
17     if (ordenada) printf("%s é ordenada\n", p);    // resultado
18     else printf("%s não é ordenada\n", p);         // resultado
19     return 0;                                     // fim sem erro
}
```

Avaliação 3 – 2014.2, Questão II

avaliacao_3_ano_2014_2.pdf

Encontros consonantais numa frase

Dicas:

- Leia a frase inteira com fgets
- Função consoante(c): letra (faixas a-z, A-Z) que não é vogal
- Conte consoantes seguidas; a 2ª do grupo forma um encontro

Para conferir: entrada 0 istmo de Stradun não fica no Brasil → a saída deve conter 3

A solução comentada está na página do curso: <https://josedihego.net/logica/#aula-25>