

Aula 27: Validação de strings

Lógica de Programação – Superior · Bloco 3

José Dihego · IFBA

O que você precisa saber hoje

Objetivo: Verificar se uma string obedece a regras, escrevendo as próprias funções de busca.

- Buscar um caractere: laço até o '`\0`'; achou, devolve 1
- `fgets` guarda o '`\n`': troque-o por '`\0`'
- Validação: começa válido; qualquer violação torna inválido
- Uma função por pergunta: `contem(s, c)`, `tira_enter(s)`

```
int contem(char s[], char c) {  
    for (int i = 0; s[i] != '\0'; i++)  
        if (s[i] == c) return 1;  
    return 0;  
}  
  
/* no main */  
int pode = 1;  
for (int i = 0; msg[i] != '\0'; i++)  
    if (!contem(alfabeto, msg[i])) pode = 0;
```

Erros que custam pontos

Errado	Certo	Por quê
<pre>fgets(...); /* e o \n? */ scanf("%d", &n); fgets(s, ...); else return 0; /* dentro do laço */</pre>	<pre>if (s[i] == '\n') s[i] = '\0'; getchar(); antes do fgets return 0; /* depois do laço */</pre>	- o Enter vira um caractere inválido - o Enter do número fica no buffer - desistiu no primeiro caractere

```
char alf[] = "!lob"; char msg[] = "obl!x!"; int
    pode = 1;
for (int i = 0; msg[i] != '\0'; i++) {
    int achou = 0;
    for (int j = 0; alf[j] != '\0'; j++) if (alf[
        j] == msg[i]) achou = 1;
    if (!achou) pode = 0;
}
```

msg[i]	está no alfabeto?	pode
o	sim	1
b	sim	1
l	sim	1
!	sim	1
x	não	0
!	sim	0

Avaliação 3 – 2025.2, Questão I

avaliacao_3_ano_2025_2.pdf

A mensagem pode ter sido escrita por alienígenas?

Estratégia:

1. Ler o alfabeto e a mensagem com `fgets` (podem ter espaços) e tirar o `Enter`
2. Cada caractere da mensagem tem que estar no alfabeto (contem)
3. Imprimir a mensagem correspondente

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

Solução (1/2)

```
1  #include <stdio.h>                                // printf e fgets
2
3  void tira_enter(char s[]) {                          // troca o '\n' que o fgets guarda por '\0'
4      for (int i = 0; s[i] != '\0'; i++)              // procura o Enter
5          if (s[i] == '\n') s[i] = '\0';              // achou: a string termina aqui
6  }
7
8  int contem(char s[], char c) {                       // 1 se o caractere c aparece em s
9      for (int i = 0; s[i] != '\0'; i++)              // percorre s
10         if (s[i] == c) return 1;                    // achou
11     return 0;                                         // chegou ao fim sem achar
12 }
```

Solução (2/2)

```
14 int main(void) { // início do programa
15     char alfabeto[100], msg[1001]; // símbolos alienígenas e mensagem interceptada
16     printf("Alfabeto: "); // pede o alfabeto
17     fgets(alfabeto, sizeof alfabeto, stdin); // lê a linha
18     tira_enter(alfabeto); // remove o Enter do fim
19     printf("Mensagem: "); // pede a mensagem
20     fgets(msg, sizeof msg, stdin); // lê a linha
21     tira_enter(msg); // remove o Enter
22     int pode = 1; // supõe que sim
23     for (int i = 0; msg[i] != '\0'; i++) // cada símbolo da mensagem
24         if (!contem(alfabeto, msg[i])) pode = 0; // símbolo fora do alfabeto
25     if (pode) printf("A mensagem pode ter sido escrita por alienígenas\n"); // nenhuma regra
        foi violada
26     else printf("A mensagem não pode ter sido escrita por alienígenas\n"); // alguma regra foi
        violada
27     return 0; // fim sem erro
28 }
```

Avaliação 3 – 2019.1, Questão I

avaliacao_3_ano_2019_1.pdf

Faixas de letras

Dicas:

- Marque as letras presentes: `tem[s[i] - 'a'] = 1`
- Faixa começa numa letra presente cuja anterior não está presente
- Estenda a faixa enquanto a próxima letra existir e imprima `início:fim`

Para conferir: entrada `iron maiden` → a saída deve conter `a:a d:e i:i m:o r:r`

A solução comentada está na página do curso: <https://josedihego.net/logica/#aula-27>