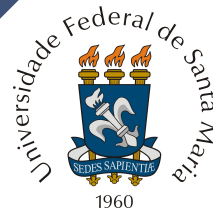


# Algoritmos


**Disciplina: Algoritmos e Programação (CPAGG101)**  
**Técnico em Informática**  
**Prof. Fernando Emilio Puntel**





## VETOR

- Deseja-se publicar o número de acertos de cada aluno em uma prova em forma de testes. A prova consta de 15 questões, cada uma com cinco alternativas identificadas por A, B, C, D e E.
- O algoritmo deve ler as respostas corretas para o gabarito que devem ser armazenadas em um vetor do tipo char.
- Após isso, o algoritmo deve ler as respostas do aluno em outro vetor do tipo char e verificar quantos acertos o aluno obteve.
  - ▶ **DICA: Utilize um espaço antes de ler o char, assim ele não irá pegar o enter**

  
`scanf(" %c", &gabarito[ind]);|`



## VETOR

- Faça um programa que carregue dois vetores de 20 posições de caracteres. A seguir, troque o 1º. elemento de A com o 20º de B, e o 2º. elemento de A, com o 19º. de B, assim por diante, até trocar o 20º. de A com o 1º. de B. Mostre os vetores antes e depois da troca.



# STRING

- String é uma variável que armazena uma cadeia de caracteres.
- Nada mais é que um vetor de char:

```
char nome[10] = {'F', 'E', 'R', 'N', 'A', 'N', 'D', 'O'};
```

F	E	R	N	A	N	D	O	\0
---	---	---	---	---	---	---	---	----



# STRING

```
1  #include <stdio.h>
2
3  int main() {
4      char nome[8];
5      int ind;
6
7      for(ind = 0 ; ind < 8; ind++){
8          scanf("%c", &nome[ind]);
9      }
10     printf("NOME DIGITADO: \n");
11     for(ind = 0 ; ind < 8; ind++){
12         printf("%c", nome[ind]);
13     }
14     return 0;
15 }
```



# STRING

## Sem string.h

```
1  #include <stdio.h>
2
3  int main(){
4      char nome[8];
5      int ind;
6
7      for(ind = 0 ; ind < 8; ind++){
8          scanf("%c", &nome[ind]);
9      }
10     printf("NOME DIGITADO: \n");
11     for(ind = 0 ; ind < 8; ind++){
12         printf("%c", nome[ind]);
13     }
14     return 0;
15 }
```

## Com string.h

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5      char nome[8];
6
7      scanf("%s", &nome);
8      printf("NOME DIGITADO: \n");
9      printf("%s", nome);
10
11     return 0;
12 }
```



# STRING

## Alguns detalhes:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char nome[8];
6
7      scanf("%s", &nome);
8      printf("NOME DIGITADO: \n");
9      printf("%s", nome);
10
11     return 0;
12 }
```

É necessário incluir a biblioteca string.h

E utilizar máscara "%s" para scanf e printf



# STRING

Outra opção é utilizar o **gets()**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    char nome[8], nome2[8];
    int resultado;

    fflush(stdin);
    gets(nome);
    fflush(stdin);
    gets(nome2);

    printf("NOME DIGITADO1: \n");
    printf("%s", nome);
    printf("\nNOME DIGITADO2: \n");
    printf("%s", nome2);

    return 0;
}
```

Contudo o **gets()** pega tudo que aparecer. Ou seja, se você der um ENTER, o **gets()** vai armazenar o enter na string. Por conta disso é necessário utilizar o **fflush(stdin);** antes do **gets()**





## STRING

O **'\0' (barra zero)** sempre será o último caracter de um vetor char para informar que o texto concluiu.

Ex. Inserir o nome Fernando em uma cadeia de caractere. O dado será armazenado da seguinte maneira:

```
char nome[10] = {'F', 'E', 'R', 'N', 'A', 'N', 'D', 'O', '\0'};
```

'F'	'E'	'R'	'N'	'A'	'N'	'D'	'O'	'\0'
-----	-----	-----	-----	-----	-----	-----	-----	------



## STRING

- Faça um programa que leia uma palavra e conte quantas letras possui a palavra



## STRING

- Faça um programa que leia uma palavra e conte quantas letras possui a palavra

```
letras = 0;
parar = 0;
while (parar == 0){
    if(nome[letras] != '\0'){
        letras++;
    }else{
        parar = 1;
    }
}
printf("TOTAL LETRAS: %i\n", letras);
```



## STRING

- Faça um programa que leia uma palavra e conte quantas vogais e consoantes possui a palavra



## STRING

- Faça um programa para criptografar uma mensagem. Onde as seguintes vogais devem ser substituídas por caracteres especiais:
  - ▷ 'A ' deve ser substituído por '@'
  - ▷ 'E' deve ser substituído por '&'
  - ▷ 'I' deve ser substituído por '1'
  - ▷ 'O' deve ser substituído por '0'
  - ▷ 'U' deve ser substituído por '#'



## STRING

- Faça um programa para criptografar uma mensagem. Onde as seguintes vogais devem ser substituídas por caracteres especiais:
  - ▷ 'A ' deve ser substituído por '@'
  - ▷ 'E' deve ser substituído por '&'
  - ▷ 'I' deve ser substituído por '1'
  - ▷ 'O' deve ser substituído por '0'
  - ▷ 'U' deve ser substituído por '#'



## STRING

- Faça um programa que receba uma frase e um caractere e verifique em que posição da frase o caractere digitado aparece pela última vez.



# STRING

- Para comparação de duas strings utiliza-se a função **strcmp()** da biblioteca **string.h**.
  - Quando o resultado for igual a 1 significa que as strings são iguais.
  - Logo, igual a zero são diferentes.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5      char nome[10], nome2[10];
6      int resultado;
7
8      // ENTRADA DOS NOMES
9      fflush(stdin);
10     printf("Digite o primeiro nome:");
11     scanf("%s", &nome);
12     fflush(stdin);
13     printf("Digite o segundo nome:");
14     scanf("%s", &nome2);
15
16     // STRCMP
17     resultado = strcmp(nome, nome2);
18     //COMPARACAO
19     if (resultado == 0){
20         printf("NOMES IGUAIS\n");
21     }else{
22         printf("NOMES DIFERENTES\n");
23     }
24     return 0;
25 }
```





## STRING

- Faça um programa que inicialmente receba uma string.
- E após isso o programa deve ler uma segunda string até que esta seja igual a primeira.



# STRING

- Também é possível utilizar uma palavra para comparar com a string. Por exemplo:
  - ▷ Verificação se o nome da pessoal é CLAUDIO

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5      char nome[10];
6      int resultado;
7
8      // ENTRADA DOS NOMES
9      fflush(stdin);
10     printf("Digite o seu nome:");
11     scanf("%s", &nome);
12
13     // STRCMP
14     resultado = strcmp(nome, "CLAUDIO");
15     //COMPARACAO
16     if (resultado == 0){
17         printf("SEU NOME EH CLAUDIO\n");
18     }else{
19         printf("SEU NOME NAO EH CLAUDIO\n");
20     }
21     return 0;
22 }
```



## STRING

- Faça um programa que leia as três primeiras letras de um mês do ano(jan, fev, mar...). Após isso, apresente o mês digitado por extenso:
  - ▷ Por exemplo:
  - ▷ Você digitou jan que representa o mês de Janeiro.



## STRING

- Faça um programa que apresente o seguinte menu:
  - ▷ Digite o destino das suas próximas férias:
    - ▷ RS
    - ▷ SC
    - ▷ PR
  - ▷ Após isso, verifique em qual estado será as férias e apresente uma mensagem de boa viagem escrevendo o nome do estado por extenso. Por exemplo:
    - ▷ Sua próxima viagem será para o Paraná.



## STRING

- Faça um algoritmo que leia e faça a média das notas de 5 alunos.
  - ▷ Cada aluno possui 3 notas que devem ser lidas e após isso fazer a média aritmética.
  - ▷ Também deve-se ler o nome do aluno.
  - ▷ Após a leitura das notas e do nome do aluno deve-se verificar se o aluno foi ou não aprovado, assumindo que a média para aprovação é 7,00.
  - ▷ Por fim deve-se mostrar uma mensagem informando a situação do aluno:
    - ▷ Ex. O aluno Fernando foi reprovado com média 5,40.
  - ▷ Após isso, repetir os passos anteriores para os demais alunos.



# Dúvidas?

Fernando Emilio Puntel  
fernandopuntel@gmail.com