

Rodrigo Henrich

rodrigohenrich@faccat.br



Vetores de palavra em C

- Já vimos que podemos usar vetores de diversos tipos de dados
- Um deles é o char, pensando desta forma podemos armazenar palavras e frases

0	1	2	3	4	5	6	7
P	A	L	A	V	R	A	

Declarando um vetor de caracteres

```
char nome[50];
```

```
char nome[50] = {'T', 'e', 's', 't', 'e'};
```

```
char palavra[10] = {'P', 'a', 'l', 'a', 'v', 'r', 'a'};
```

Declarando um vetor de caracteres

```
4 int main(){
5     setlocale(LC_ALL, "");
6     setbuf(stdin, NULL);
7     char palavra[10] = {'P', 'a', 'l', 'a', 'v', 'r', 'a'};
8     //Segunda forma de declarar e inicializar um vetor de caracteres
9     char nome[30] = "Rodrigo Henrich";
10    //Desta forma a capacidade do vetor será calculada automaticamente
11    char disciplina[] = "Programação";
12
13 }
```

Strings

- Strings em c, são vetores de caracteres
- Para marcar o fim de uma String em C o último caractere sempre é o `\0`
- Ele será inserido automaticamente durante a leitura de uma String do teclado;
- Importante lembrar de sempre declarar um espaço a mais para o `\0`

0	1	2	3	4	5	6	7	8	9
P	A	L	A	V	R	A	\0		

Mostrando uma String na tela

- Para mostrar uma String na tela podemos usar o %s dentro do printf

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "");
6      setbuf(stdin, NULL);
7      char nome[20] = "Programação C";
8      printf("%s", nome);
9  }
```

Lendo string em C

- Podemos usar a função **scanf** para ler uma string, no entanto ela serve para ler apenas uma palavra
- Ele lê até encontrar um espaço, então não serve para ler frases
- Note que para leitura de Strings não é preciso o **&** antes do nome da variável

```
char nome[20];  
printf("Digite seu nome: ");  
scanf("%s", nome);
```

Lendo string em C

- Outra opção é usar a função **gets()**, ela armazena tudo o que foi digitado, inclusive os espaços.

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      setbuf(stdin, NULL);
6      char nome[20];
7      printf("Digite seu nome: ");
8      gets(nome);
9  }
```


Percorrendo um vetor de caracteres

- Para percorrer podemos percorrer ele da mesma forma que fazemos com outros vetores

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      int i;
5      setlocale(LC_ALL,"");
6      setbuf(stdin,NULL);
7      char nome[] = "Palavra";
8      for(i=0;i<7;i++)
9          printf("\n[%d] - %c",i,nome[i]);
10 }
```

Percorrendo um vetor de caracteres

- Outra forma é percorrer até encontrar o `'\0'` que marca o final de um vetor de caractere

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      int i;
5      setlocale(LC_ALL,"");
6      setbuf(stdin,NULL);
7      char nome[] = "Palavra";
8      for(i=0; nome[i] != '\0'; i++)
9          printf("\n[%d] - %c", i, nome[i]);
10 }
```

Saber o tamanho de um vetor de caracteres

- Podemos percorrer até encontrar o `\0` para saber o tamanho de uma string

Saber o tamanho de um vetor de caracteres

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      int tamanho;
5      setlocale(LC_ALL, "");
6      setbuf(stdin, NULL);
7      char nome[] = "Palavra";
8      for(tamanho=0; nome[tamanho] != '\0'; tamanho++);
9      printf("A string '%s' tem %d caracteres.", nome, tamanho);
10 }
```