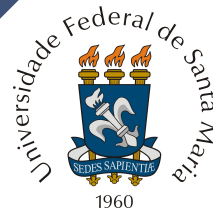


Algoritmos

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



Passagem por **valor**

```
1  #include <stdio.h>
2
3  void alteraIdade(int idade);
4
5  int main() {
6      int idade = 10;
7
8      printf("Idade: %i\n", idade);
9      alteraIdade(idade);
10     printf("Idade: %i\n", idade);
11
12
13     return 0;
14 }
15
16 void alteraIdade(int idade) {
17     idade = idade + 5;
18 }
```



```
C:\...
Idade: 10
Idade: 10
Process returned
Press any key to
```

Passagem por **referência**

Passagem por **valor**

```
1  #include <stdio.h>
2
3  void alteraIdade(int idade);
4
5  int main() {
6      int idade = 10;
7
8      printf("Idade: %i\n", idade);
9      alteraIdade(idade);
10     printf("Idade: %i\n", idade);
11
12
13     return 0;
14 }
15
16 void alteraIdade(int idade) {
17     idade = idade + 5;
18 }
```



```
C:\... - [icon] x
Idade: 10
Idade: 10
Process returned 0
Press any key to
```

Passagem por **referência**

```
1  #include <stdio.h>
2
3  void alteraIdade(int *idade);
4
5  int main() {
6      int idade = 10;
7
8      printf("Idade: %i\n", idade);
9      alteraIdade(&idade);
10     printf("Idade: %i\n", idade);
11
12
13     return 0;
14 }
15
16 void alteraIdade(int *idade) {
17     *idade = *idade + 5;
18 }
19
```



```
C:\T... - [icon] x
Idade: 10
Idade: 15
Process returned 0
Press any key to c
```

Passagem por valor

```
1 #include <stdio.h>
2
3 void alteraIdade(int idade);
```

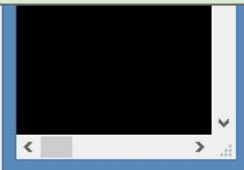


Idade: 10

Neste caso:

- O valor alterado dentro da função alteraIdade **NÃO IRÁ** alterar diretamente o valor da variável na função int main().

```
13 return 0;
14 }
15
16 void alteraIdade(int idade){
17     idade = idade + 5;
18 }
```



Passagem por referência

```
1 #include <stdio.h>
2
3 void alteraIdade(int *idade);
```

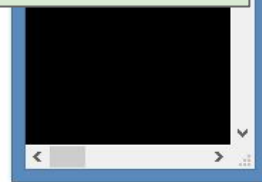


Idade: 10

Neste caso:

- O valor alterado dentro da função alteraIdade **IRÁ** alterar diretamente o valor da variável na função int main().

```
13 return 0;
14 }
15
16 void alteraIdade(int *idade){
17     *idade = *idade + 5;
18 }
19
```



Passagem por valor



```
alteraIdade(idade);
```

Passagem por referência



```
alteraIdade(&idade);
```

Chamada da função:

Função:

```
15  
16 void alteraIdade(int idade) {  
17     idade = idade + 5;  
18 }
```

```
15  
16 void alteraIdade(int *idade) {  
17     *idade = *idade + 5;  
18 }  
19
```

Printf após a chamada da função:

```
C:\...  
Idade: 10  
Idade: 10
```

```
C:\T...  
Idade: 10  
Idade: 15
```



PASSAGEM DE PARÂMETRO POR REFERÊNCIA

- Ex.
 - ▷ Faça um algoritmo que leia duas notas de dois alunos e através de uma função realize a média das notas.
 - ▷ A variável média deve ser passada por referência na função.

PASSAGEM DE PARÂMETRO POR REFERÊNCIA

Passagem de valor por referência são utilizados em casos onde desejamos criar uma função que retorne **mais de um valor**.

Quando utilizamos uma função com esse escopo:

```
float notas (float n1, float n2, float media, char situacao){
```

Sabemos que a função só irá retornar **UM** valor do tipo **float**.

Porém se utilizarmos uma função com esse escopo:

```
float notas (float n1, float n2, float *media, char *situacao){
```

Além do retorno de uma variável float, os dados alterados nas variáveis **media** e **situacao** alterados na função também serão “alterados” nas variáveis do main.



ATIVIDADE

- Escreva um programa em C que leia uma string e imprima quantas consoantes e vogais existem na string. O programa deve ser implementado com a utilização de uma função.
- Passe as variáveis que contabilizam o total de vogais e consoantes como referência.



ATIVIDADE

- Escreva um programa em C para ler 5 **pares** de valores (considere que serão informados apenas valores positivos).
- Para cada par lido deve ser impresso o valor do maior elemento do par ou a frase "Eles são iguais" se os valores do par forem iguais.
- Para obter os números lidos utilize uma função onde os valores serão passados como referência.
- Para obter o maior elemento do par utilize uma função.

- Escreva uma função para calcular a nota final de um aluno. A função deve receber três notas e duas letras:
 - ▷ Uma letra em relação ao tipo de média que será realizado: se a letra for **A** a função retorna a média aritmética das notas do aluno, se for **P**, a sua média ponderada (pesos: 5, 3 e 2).
 - ▷ A outra letra correspondente a nota do aluno:
 - ▷ A: entre 10 e 8.
 - ▷ B: entre 8 e 6.
 - ▷ C: entre 6 e 4.
 - ▷ D: entre 4 e 0.
 - ▷ Após a realização da média a função deve retornar para o main onde deve-se apresentar uma mensagem informando a situação do aluno. Por exemplo:
 - ▷ Média: 7.45. Situação: A.



ATIVIDADE

- Escreva um programa que leia 5 números inteiros positivos (utilize uma função que leia esse número e verifique se ele é positivo).
- Para cada número informado escrever a soma de seus divisores (exceto ele mesmo).
 - ▷ Crie uma função para obter a soma.
 - ▷ Entrada: Um número inteiro e positivo.
 - ▷ Saída: A soma dos divisores.
 - ▷ Exemplo: Para o valor 8: $1+2+4 = 7$

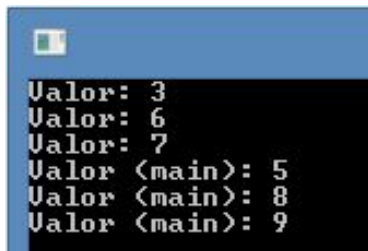


PASSAGEM DE UM VETOR POR PARÂMETRO

```
1  #include <stdio.h>
2
3  void recebeVetor(int vetor[]);
4
5  int main() {
6      int vetor[3] = {3, 6, 7};
7
8      recebeVetor(vetor);
9
10     return 0;
11 }
12
13 void recebeVetor(int vetor[]) {
14     int ind;
15     for(ind = 0; ind < 3; ind++) {
16         printf("Valor: %i\n", vetor[ind]);
17     }
18 }
```

VETOR

Todas as alterações realizadas no vetor na função irão impactar diretamente no vetor do int.



```
Valor: 3
Valor: 6
Valor: 7
Valor (main): 5
Valor (main): 8
Valor (main): 9
```

```
1  #include <stdio.h>
2
3  void recebeVetor(int vetor[]);
4
5  int main() {
6      int vetor[3] = {3, 6, 7}, ind;
7
8      recebeVetor(vetor);
9      for(ind = 0; ind < 3; ind++){
10         printf("Valor (main): %i\n", vetor[ind]);
11         vetor[ind] = vetor[ind] + 2;
12     }
13     return 0;
14 }
15
16 void recebeVetor(int vetor[]){
17     int ind;
18     for(ind = 0; ind < 3; ind++){
19         printf("Valor: %i\n", vetor[ind]);
20         vetor[ind] = vetor[ind] + 2;
21     }
22 }
```



ATIVIDADE

- Faça uma função em C que recebe um vetor de 10 posições como parâmetro. A função deve alterar a ordem dos valores no vetor, por exemplo:
 - ▷ posição 1 <-> posição 10
- Ao final o algoritmo deve apresentar o vetor resultante no main.



ATIVIDADE

- Escreva um programa em C que leia as 3 medidas dos lados de 5 triângulos. Para cada triângulo imprimir a sua classificação:
 - ▷ Triângulo Equilátero (3 lados iguais).
 - ▷ Triângulo Isósceles (2 lados iguais).
 - ▷ Triângulo Escaleno (3 lados diferentes).
- O programa deve aceitar apenas valores positivos para as medidas dos lados e deve possuir duas funções:
 - ▷ Uma para obter os valores dos lados (passagem de parâmetros por referência).
 - ▷ Outra para verificar o vetor e apresentar a classificação dos triângulos.

ATIVIDADE

- Um mercado quer analisar quantas transações realizadas no último mês foram de entrada (E) e quantas foram de saída (S) e o valor disponível em caixa (considere que o caixa está zerado no início do mês). Para armazenamento dos valores transações utilize um vetor (float) e para que tipo de transações utilize outro vetor (char).
- Com isso, faça um algoritmo que leia o valor de 30 transações e classifique as transações no vetor do tipo de transação.
- Utilize duas funções:
 - ▷ Uma função que leia e retorne o valor da transação.
 - ▷ Uma função que recebe o valor da transação e retorna a classificação da transação.
- Ao final apresente os valores das transações e o tipo dela, por exemplo:
 - ▷ Transação: 200 E <- Neste caso teve R\$200,00 de entrada
 - ▷ Transação: -15 S <- Neste caso teve R\$15,00 de saída
- Também apresente o total em caixa do mercado.

ATIVIDADE

- Uma loja que vende CD-ROM deseja automatizar o controle de suas vendas. Escreva um programa em C que controle a venda diária dessa loja da seguinte forma. A cada compra efetuada por um cliente deve ser informado o código do vendedor (codificado da seguinte forma: **1.Silva 2.Soures 3.Sair**) que efetuou a venda e a respectiva quantidade de CDs vendidos.
- Após os dados de entrada, os totais devem ser atualizados e retornar para um novo movimento. Se for informado o código 3 (sair) o programa deverá encerrar imediatamente imprimindo um resumo do dia com as seguintes informações: Número de atendimentos do Silva, número de atendimentos do Soares, número total de CD-ROMs vendidos e uma mensagem indicando a situação do dia conforme a mensagem abaixo:
 - ▷ Abaixo de 10 vendas: Dia péssimo
 - ▷ Acima de 50 vendas: Dia ótimo
 - ▷ Qualquer outro valor: Dia normal
- Utilize uma função para realizar a soma do total de CDs vendidos para cada funcionário e o total da loja. A função deve receber o código do funcionário, total de DVDs vendido e total de DVDs vendido no dia da loja. A função não deve ter retorno. (referência)



Dúvidas?

Fernando Emilio Puntel
fernandopuntel@gmail.com