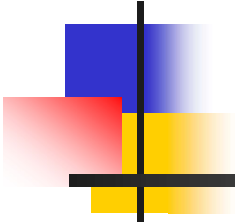


UNIVERSIDADE ESTADUAL DE MS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

A decorative graphic on the left side of the slide consists of a black crosshair. The vertical line of the crosshair passes through a blue square, a red square, and a yellow square. The horizontal line of the crosshair passes through the same three squares.

Disciplina: AEDI

Modularização

parte 3

Professor: Nilton

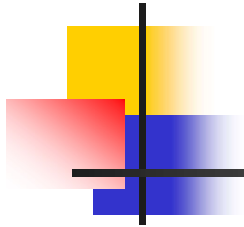
E-mail: nilton@comp.uems.br



Tipos de parâmetros

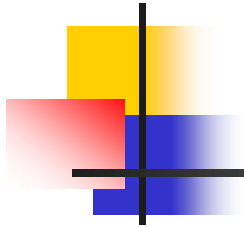
- Passagem por valor
- Passagem por referência

Passando parâmetros por valor



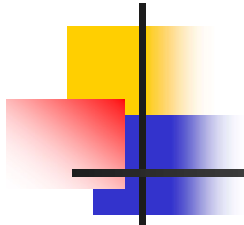
- Um **parâmetro por valor** recebe a **cópia** da variável existente na chamada da rotina.
- Qualquer modificação do conteúdo do parâmetro dentro da rotina **não é "enxergado"** fora da rotina através da variável existente na chamada da rotina.

Passando parâmetros por referência



- Um **parâmetro por referência** recebe o **endereço** da variável existente na chamada da rotina.
- Qualquer modificação do conteúdo do parâmetro dentro da rotina **é "enxergado"** fora da rotina através da variável existente na chamada da rotina.
- Para um parâmetro armazenar um endereço de uma outra variável, devemos declarar o parâmetro como **ponteiro**.

Ponteiro



- Se temos a declaração

```
int *p;      //declaração de um ponteiro para o tipo inteiro
```

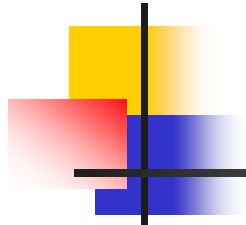
Então, **&p** → obtém o endereço de p
 ***p** → manipula o conteúdo apontado por p

```
int X = 0;  
int *A;
```

```
A = &X;      // A passa a “apontar” para X
```

```
*A = 100;    // X recebe o conteúdo 100
```

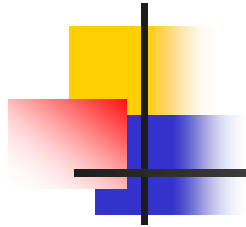
Dentre dois números obter o maior e o menor – impl1



```
void MaiorMenor(int a, int b, int *maior, int *menor) {  
    if(a>b) {  
        *maior=a;  
        *menor=b;  
    } else {  
        *maior=b;  
        *menor=a;  
    }  
}
```

```
int main(void)  
{  
    int ma, me, x=20, y=5;  
  
    MaiorMenor(2,10,&ma,&me);  
    printf("\n%d %d",ma,me);  
    MaiorMenor(x,y,&ma,&me);  
    printf("\n%d %d",ma,me);  
    return 0;  
}
```

Dentre dois números obter o maior e o menor – impl2

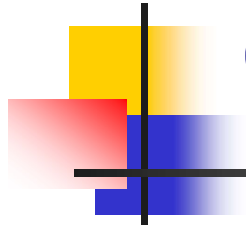


```
void MaiorMenor(int *maior, int *menor) {  
    int aux;
```

```
    if(*menor>*maior) {  
        aux=*maior;  
        *maior=*menor;  
        *menor=aux;  
    }  
}
```

```
int main(void)  
{  
    int ma=2, me=10;  
  
    MaiorMenor(&ma,&me);  
    printf("\n%d %d",ma,me);  
    return 0;  
}
```

Colocar 3 valores em ordem crescente



```
void MenorMaior(int *menor, int *maior) {
    int aux;

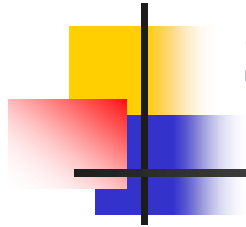
    if(*menor>*maior) {
        aux=*maior;
        *maior=*menor;
        *menor=aux;
    }
}

void OrdenaCrescente(int *n1, int *n2, int *n3) {
    MenorMaior(n1,n2);
    MenorMaior(n1,n3);
    MenorMaior(n2,n3);
}

int main(void)
{
    int menor=3, meio=2, maior=1;

    OrdenaCrescente(&menor,&meio,&maior);
    printf("\n%d %d %d",menor,meio,
        maior);
    return 0;
}
```


Encontrar o dobro de um número se for par e o triplo se for ímpar

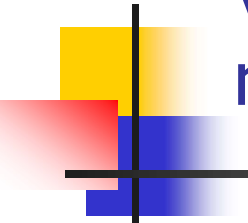


```
void DobroTriplo(int n, int *d, int *t) {
```

```
    if(n % 2 == 0) {
        *d=n*2;
        *t=0;
    } else {
        *d=0;
        *t=n*3;
    }
}
```

```
int main(void)
{
    int numero=2, dobro, triplo;

    DobroTriplo(numero,&dobro,&triplo);
    printf("\n%d %d %d",numero,dobro,
        triplo);
    return 0;
}
```

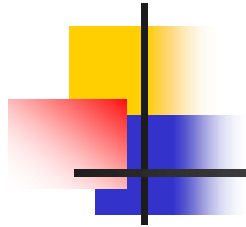


Para um número de votos brancos, nulos e válidos, determinar o percentual que cada um representa em relação ao total de eleitores

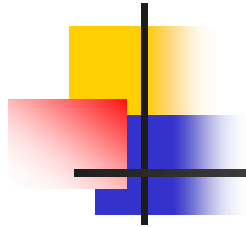
```
void PercVotos(int nvotosbrancos, int nvotosnulos, int nvotosvalidos,  
               int *percbrancos, int *percnulos, int *percvalidos) {
```

```
    // implemente o código
```

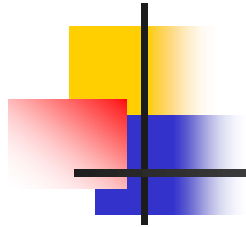
```
}
```



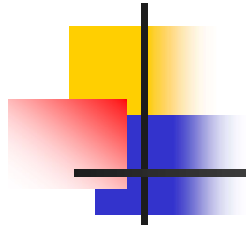
Dado um número, determine se ele é divisível por 10, por 5, por 2, ou se não é divisível por nenhum deles.



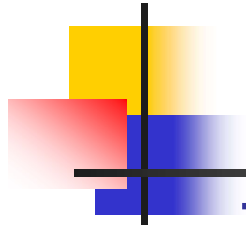
João recebeu seu salário e deve duas contas que estão atrasadas. Como as contas estão atrasadas, ele terá que pagar multa de 2% sobre cada conta. Determinar quais contas poderão ser pagas e quanto restará de seu salário.



A lanchonete Gostosura vende apenas um tipo de sanduíche, cujo recheio inclui duas fatias de queijo, uma fatia de presunto e uma rodela de hambúrguer. Sabendo que cada fatia de queijo ou presunto pesa 50 gramas e que a rodela de hambúrguer pesa 100 gramas, informe as quantidades (em quilos) de queijo, presunto e carne necessários para compra, fornecido a quantidade de sanduíches a fazer pelo dono.



Dadas quatro notas, informá-las em ordem crescente.



Transforme segundos em horas, minutos e segundos.

Passagem de parâmetros de dados homogêneos



- Por padrão, quando se passa um vetor/matriz por parâmetro, a linguagem C interpreta como passagem por referência.


```
#include <cstdlib>
#include <stdio.h>

void inicilizaVetor(int V[], int n)
{
    for(int i=0;i<n;i++) {
        V[i]=1;
    }
}

int main(int argc, char *argv[])
{
    int Vetor[10];

    printf("\nvalores do vetor antes da inicializacao\n");
    for(int i=0;i<10;i++)
        printf("%d ",Vetor[i]);

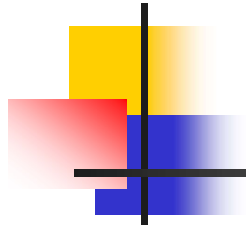
    printf("\n\nvalores do vetor apos a inicializacao\n");
    inicilizaVetor(Vetor,10);
    for(int i=0;i<10;i++)
        printf("%d ",Vetor[i]);

    printf("\n\n");

    system("PAUSE");
    return 0;
}
```



```
C:\Dev-Cpp\main_5.exe
valores do vetor antes da inicializacao
2293600 2009067269 2293648 4198456 42024
valores do vetor apos a inicializacao
1 1 1 1 1 1 1 1 1 1
Pressione qualquer tecla para continuar.
```



```
void inicilizaVetor(int *V, int n)
{
    for(int i=0;i<n;i++) {
        V[i]=1;
    }
}
```



Exercícios

Resolva as listas anteriores usando rotinas com passagem de parâmetros por valor e/ou referência.