

Algoritmos e Programação

Matrizes e Funções em C

Anderson Fortes





Aula passada...

Expressões reduzidas e laços de repetição.

Expressões reduzidas

Expressão	Forma compacta
$x = x + y$	$x += y$
$x = x - y$	$x -= y$
$x = x * y$	$x *= y$
$x = x / y$	$x /= y$
$x = x \% y$	$x \% = y$

Se uma expressão incrementa ou decrementa o valor da variável, podemos então escrevê-la numa forma ainda mais compacta.

ex: $x = x + 1$

$x++$, $x--$

Podemos escrever como

$x++$

Laço de repetição for

for(inicialização; condição; alteração)

Ex: Mostrar os números de 10 até 100.

```
int main() {  
    int i;  
  
    for (i=10; i<=100; i++){  
        printf("%d \n",i);  
    }  
    return 0;  
}
```

Laço de repetição while

while(condição) comando;

Ex: Mostrar os números de 10 até 100.

```
#include<stdio.h>

int main(){

    int i;

    i=0;

    while (i<=100){
        printf("%i\n",i);
        i++;
    }
}
```

Hoje

Funções em C



para que as funções servem mesmo?

Declarando uma função



Tipo do retorno

```
int maior (int a, int b){  
    if (a>b)  
        return a;  
    else  
        return b;  
}
```

Declarando uma função



Tipo do retorno

Nome da função

```
int maior (int a, int b){  
    if (a>b)  
        return a;  
    else  
        return b;  
}
```


Declarando uma função



Tipo do retorno

Nome da função

```
int maior (int a, int b){  
    if (a>b)  
        return a;  
    else  
        return b;  
}
```

Parâmetros

Declarando uma função



```
int maior (int a, int b){  
    if (a>b)  
        return a;  
    else  
        return b;  
}
```

```
funcao maior (a,b:inteiro):inteiro  
inicio  
    se a> b entao  
        retorne a  
    senao  
        retorne b  
    fimse  
fimfuncao
```

Declarando uma função



```
int maior (int a, int b){  
    if (a>b)  
        return a;  
    else  
        return b;  
}
```

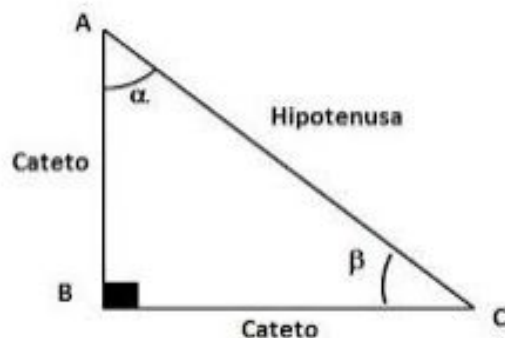
```
int main(){  
    int n1 = 20;  
    int n2 = 40;  
  
    printf("O maior entre %d e %d eh %d", a, b, maior(n1,n2));  
  
    return 0;  
}
```

Exemplo:



9 - Crie uma função que receba a comprimento dos catetos de um triângulo-retângulo e calcule e retorne o comprimento da hipotenusa, sabendo que

$$h^2 = C1^2 + C2^2$$



Exemplo:



9 - Crie uma função que receba a comprimento dos catetos de um triângulo-retângulo e calcule e retorne o comprimento da hipotenusa, sabendo que

```
float hipotenusa (float c1, float c2){  
    float h;  
    h = sqrt((c1*c1)+(c2*c2));  
  
    return h;  
}
```

E função que não retorna nada?



No visualg tínhamos os procedimentos, que não tinham nenhum tipo de retorno.

```
procedimento imprime_cabecalho (nome: caracter)  
inicio  
  
    escreval("Bem vindo, ", nome)  
fimprocedimento
```

E função que não retorna nada?



No visualg tínhamos os procedimentos, que não tinham nenhum tipo de retorno.

```
procedimento imprime_cabecalho (nome: caracter)  
inicio  
  
    escreval("Bem vindo, ", nome)  
fimprocedimento
```

```
void imprime_cabecalho (char nome[10]){  
  
    printf("Ola %s", nome);  
}
```

Importando nossas funções



Podemos salvar nossas funções e importar nos códigos que queremos usar.

nome que salvei

minhas_funcoes.c

```
2  #include<math.h>
3
4  float hip(float c1, float c2){
5      float h;
6
7      h = sqrt(pow(c1,2)+pow(c2,2));
8      return h;
9  }
```

```
#include<stdio.h>
#include"minhas_funcoes.c"

int main(){
    float c1,c2;
    printf("Digite o cateto 1: ");
    scanf("%f",&c1);

    printf("Digite o cateto 2: ");
    scanf("%f",&c2);

    printf("A hipotenusa vale %.2f", hip(c1,c2));

    return 0;
}
```


Onde declarar a função?



Não é necessário declarar sempre acima da função main, desde que na main façamos uma declaração de quais funções existem.

```
int main(){  
    void imprime(int num);  
    imprime(5);  
}  
  
void imprime(int num){  
    printf("Numero digitado: %i", numero);  
}
```

para poder utilizar a função que é declarada abaixo da main, obrigatoriamente tenho que declarar ela como uma variável.

Exemplos



1 - Criar uma função chamada `isPositive` que recebe um número e retorne se esse número é positivo ou não (VERDADEIRO ou FALSO).

2 - Crie uma função chamada `Dolar2Real` (Dolar to Real) que recebe um valor em dólares e retorna o valor em real. Considere que U\$\$ 1,00 = R\$ 5.50.

3 - Codifique a função `fat(n)`, que **retorna** o fatorial de n .

4 - Escreva uma função chamada `somaIntervalo` que recebe do programa principal 2 números inteiros $n1$ e $n2$ como parâmetro e retorna a soma de todos os números inteiros contidos no intervalo $[n1, n2]$.

5 - Crie uma função `tabuada(n)` que **mostra** a tabuada de n .