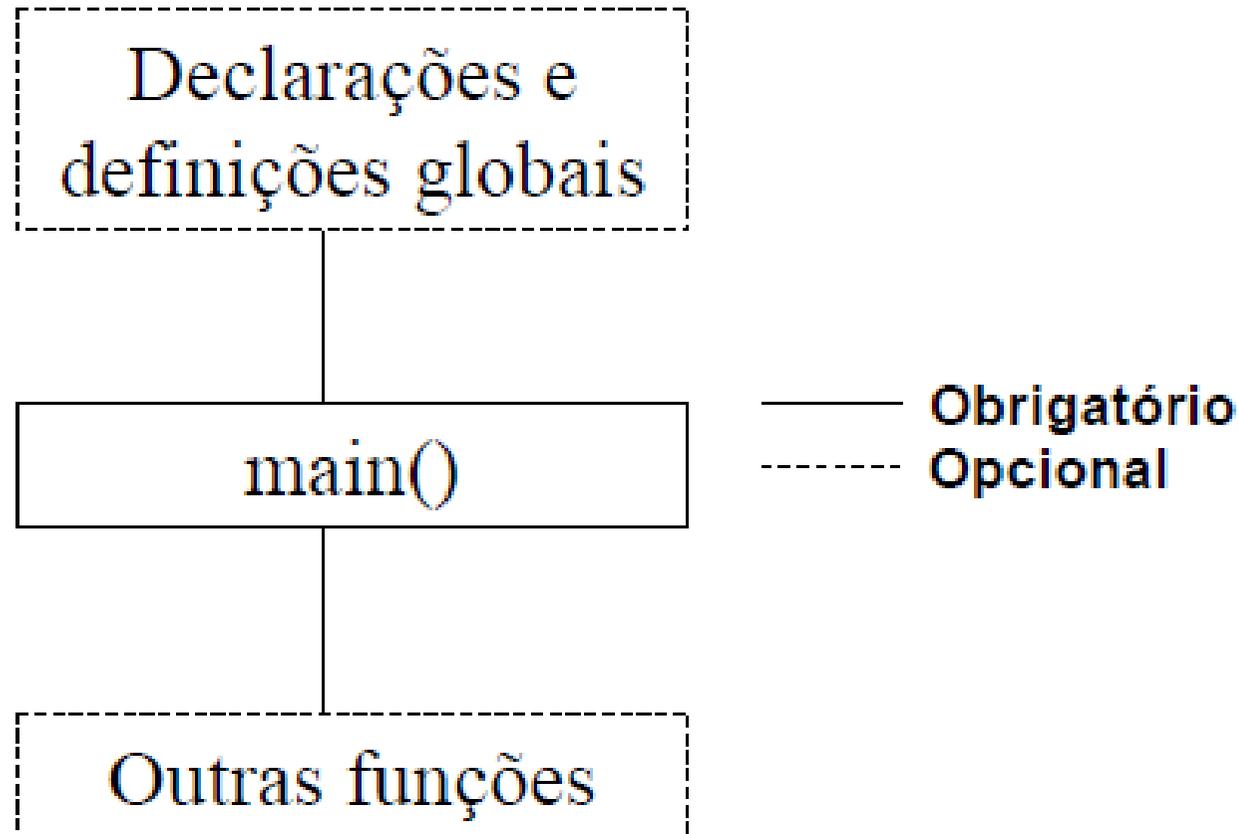


Algoritmos e Programação – Parte 02

Introdução ao C



Estrutura de um programa em C



Estrutura de um programa em C



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

```
/* Imprime a mensagem Olá Mundo! na sua tela! */
```

```
int main()  
{  
    printf ("Ola Mundo!");  
    return (0);  
}
```

Definição de Variáveis

- Devem ser declaradas no **início** do programa ou do sub bloco;
- Podem ser classificadas como **Locais** ou **Globais**.

□ Locais

- Declaradas dentro de funções;
- Utilizada apenas dentro do escopo da função;
- O escopo de uma função é determinado por abre-chaves “{“ e termina em fecha-chaves “}”;
- **Só existem** no momento que sua função está em execução.

□ Globais

- Declaradas **fora** de todas as funções;
- Podem ser **acessadas de qualquer parte** do programa;
- **Existem durante toda a execução do programa.**

Nomes de Variáveis

- Deve conter um ou mais caracteres;
- O primeiro caractere **sempre** deve ser uma **letra**;
- Os caracteres **subseqüentes** podem ser **letras**, **números** ou “_”;
- Não pode ser igual às **palavras-chaves**;
- Não pode ter o **mesmo nome de funções**;

Correto	Incorreto
Soma1	1soma
Soma	soma!
area_triangulo	area...triangulo

Obs: as variáveis “**soma**” e “**Soma**” são **distintas**

Declarando variáveis

Sintaxe

- **<Tipo de dados> Nome_variável;**

Ex:

```
char nome;
```

```
int idade;
```

```
int total;
```

Atribuindo valor

- **Nome_da_variavel = expressão;**

Ex:

```
nome = 'Joao' ;
```

```
idade = 18;
```

```
total = 10 + 20;
```

Isto é importante pois quando o C cria uma variável ele *não* a inicializa. Isto significa que até que um primeiro valor seja atribuído à nova variável ela tem um valor *indefinido* e que não pode ser utilizado para nada. *Nunca* presume que uma variável declarada vale zero ou qualquer outro valor.

Operadores aritméticos

Operador Binário	Descrição
=	Atribuição
+	Soma
-	Subtração
/	Divisão
%	Módulo (resto da divisão)

Operadores aritméticos Unários e Binários

- **Unários** (+, -, ++, --) agem sobre **uma variável** apenas, **modificando ou não** o seu valor, e retornam o valor final da variável.
 - **a = -b;**
 - **a++;** (ou seja) **a = a+1;**
 - **a--;** (ou seja) **a = a-1;**

Obs: operador “-” como troca de sinal é um operador unário que não altera a variável sobre a qual é aplicado, pois ele retorna o valor da variável multiplicado por -1.
- **Binários** (+, -, *, /, %) usam **duas variáveis** e retornam um terceiro valor, **sem modificar as variáveis originais.**

Operadores aritméticos - Hierarquia

Hierarquia	Operação
1	Parênteses
2	Função
3	++, --
4	- (menos unário)
5	*, /, %
6	+, -

Operadores de Atribuição

`=, +=, -=, *=, /=, %=`

Instrução normal	Instrução reduzida
<code>var = var + expr;</code>	<code>var += expr;</code>
<code>var = var - expr;</code>	<code>var -= expr;</code>
<code>var = var / expr;</code>	<code>var /= expr;</code>
<code>var = var * expr;</code>	<code>var *= expr;</code>

□ Exemplos:

- `a = 5;`
- `a += 5;` (ou seja) `a = (a + 5);`
- `a -= 5;` (ou seja) `a = (a - 5);`

Comentários



- `//` Meu comentário em uma linha
- `/*` Meu comentário através de um bloco de texto que pode estar em n linhas `*/`

Tipos Primitivos

Void

- Tipo de dado sem valor.

Caractere

- Definido pela palavra reservada **char**;
- Exemplo:
 - **char** letra;
 - letra = 'A' ;

Inteiro

- Definido pela palavra reservada **int**;
- Exemplo:
 - int num;
 - num = -73;

Tipos Primitivos

Ponto flutuante

- Definido pela palavra reservada **float**
- Ocupa 4 bytes
- Exemplo:
 - **float** a,b,c=2.34;

Ponto flutuante de precisão dupla

- Definido pela palavra reservada **double**
- Ocupa 8 bytes
- Exemplo:
 - **double** x=2.38, y=3.1415;

Operadores Relacionais



Operador	Ação
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual
==	igual a
!=	diferente de



Operadores Lógicos

Operador	Ação
&&	E
 	OU
!	Não

		Não p	Não q	p E q	p OU q
p	q	! p	! q	p && q	p q
falso	falso	verdadeiro	verdadeiro	falso	falso
falso	verdadeiro	Verdadeiro	falso	falso	verdadeiro
verdadeiro	falso	falso	verdadeiro	falso	verdadeiro
verdadeiro	verdadeiro	falso	falso	verdadeiro	verdadeiro

Hierarquia dos Operadores Relacionais e Lógicos

- ❑ Hierarquia ou Precedência – prioridade com que os operadores são executados pelo compilador;
- ❑ Operadores com mesmo nível hierárquico são executados da esquerda para a direita;
- ❑ Podem ser alterada utilizando “()”.

Hierarquia	Operação
1	!
2	>, >=, <, <=
3	==, !=
4	&&
5	

Funções de Entrada e Saída Formatada

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

io → input/output

Forma geral da saída:

- `printf(string_de_controle, <lista_de_argumentos>);`
- Indica as variáveis com suas respectivas posições através dos códigos de formato “%” mostrado a seguir.

Funções de Entrada e Saída Formatada

Exemplos:

- `printf ("%d", Y);` //Mostra o n° inteiro armazenado em Y.
- `printf("Conteúdo de Y = %d", Y);` //mostra a mensagem Conteúdo de Y = e seguida o n° inteiro armazenado em Y.
- `printf ("%f", X);` //Mostra o n° real armazenado em X.
- `printf ("%5.2f", X);` //Mostra o n° real armazenado em X, utilizando 5 casas para parte inteira e 2 casas para a decimal.

Código	Imprime
<code>printf ("Um %%%c indica %s",'c',"char");</code>	Um %c indica char
<code>printf ("%X %f %e",107,49.67,49.67);</code>	6B 49.670000 4.967000e+001
<code>printf ("%d %o",10,10);</code>	10 12

Funções de Entrada e Saída - `scanf()`

`scanf()` - Leitura de dados

Sintaxe:

`scanf(string_de_controle, lista_de_argumentos);`

- **string_de_controle** → descrição de todas as variáveis que serão lidas, com informações de seus tipos e da ordem em que serão lidas;
- **lista_de_argumentos** → lista com os identificadores das variáveis que serão lidas.
- Importante: colocar antes de cada variável da lista_de_argumentos o caracter '&'

Exemplo:

```
char letra; //Declarando a variável "letra"
```

```
scanf("%c", &letra); //Lendo dados digitados pelo usuário
```

Funções de Entrada e Saída

- Tabela de códigos de **tipos** de dados

Código	Formato
%c	Um caracter (char)
%d	Um número inteiro decimal (int)
%f	Ponto flutuante decimal
%s	String

Exemplos:

```
char letra;  
float nota;  
int quantDeFilhos;  
scanf("%c", &letra);  
scanf("%f", &nota);  
scanf("%d", &quantDeFilhos);
```

Funções de Entrada e Saída Formatada

Exercício:

Construa um programa que escreva a string "Juros de", o inteiro 10 e o caractere '%' na tela, exibindo a seguinte frase: Juros de 10%

~~Seria assim a resposta?~~

```
#include <stdio.h>
main ()
{
printf ("Juros de 10%");
}
```

```
#include <stdio.h>
main ()
{
printf("%s %d %c", "juros de", 10, '%');
}
```

Algoritmos e Programação

Exemplo 01 - Programa básico em C

```
#include <stdio.h> // informa ao compilador que ele deve incluir os arquivo-
#include <conio.h> //cabecalhos stdio.h e conio.h

int main () //Função principal obrigatória. INT informa que seu retorno é
           //um inteiro
{
    printf ("Meu primeiro \n programa em 'C!' \n"); // "\n" quebra de linha
    getch(); //Recurso necessário para manter aberta a janela no //
WINDOWS
    return(0); //Valor retornado pela função "INT MAIN()"
}
```

Algoritmos e Programação

Exemplo 02

/* Programa Soma de dois números

Entrada: dois numeros

Saída: soma */

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

```
#include <conio.h>
```

```
int main() {
```

```
    int a,b,c;
```

```
    printf ("Digite dois numeros\n");
```

```
    scanf ("%d",&a);
```

```
    scanf ("%d",&b);
```

```
    c=a+b;
```

```
    printf ("A soma eh: %d \t",c);
```

```
    getch();
```

```
    return (0);
```

```
}
```

Algoritmos e Programação

Exemplo 03

// Programa Cálculo da média de 02 notas

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

```
#include <conio.h>
```

```
int main() {  
    int a,b;  
    float c;  
    printf("Digite duas notas\n");  
    scanf("%d" "%d",&a,&b);  
    c=(a+b)/2;  
    printf("A media eh: %.2f",c);  
    getch();  
    return (0); }
```

Algoritmos e Programação

Exemplo 04

Construa um programa em C que receba três notas e seus respectivos pesos, calcule e mostre a média ponderada dessas notas.

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

int main() {
    float n1,n2,n3,p1,p2,p3,med;
    printf("Digite três medias:\n");
    scanf("%2f" "%2f" "%2f",&n1,&n2,&n3);
    printf("Digite o peso de cada media:\n");
    scanf("%f" "%f" "%f",&p1,&p2,&p3);
    med=((n1*p1)+(n2*p2)+(n3*p3))/(p1+p2+p3);
    printf("A media eh: %f" ,med);
    getch();
    return (0); }
```

Algoritmos e Programação

Exemplo 05

Faça um programa que leia uma temperatura em graus Celsius e calcule o correspondente em Fahrenheit. Sabendo que:
 $F = (180 * (C + 32)) / 100$.

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

```
int main() {
    float c,fa;
    printf("Digite uma temperatura em Graus Celsius:\n");
    scanf("%f" ,&c);
    fa=(180*(c+32))/100;
    printf("A temperatura correspondente em Fahrenheit eh: %f" ,fa);
    getch();
    return (0); }
```