

Tópico 2

Linguagem C

Em 1950, um grupo de programadores da IBM liderados por John Backus produz a versão inicial da linguagem FORTRAN (FORMula TRANSform). Foi a primeira linguagem de alto nível a ser criada, nessa onda várias outras linguagens de alto nível foram criadas Algol-60, Cobol, Pascal, etc.

Uma das mais bem sucedidas foi uma linguagem chamada C criada em 1972 nos laboratórios por Dennis Ritchie. Foi revisada e padronizada pela ANSI em 1989.

OBS*: ANSI: American National Standards Institute Padrão mais utilizado.

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

int main(){

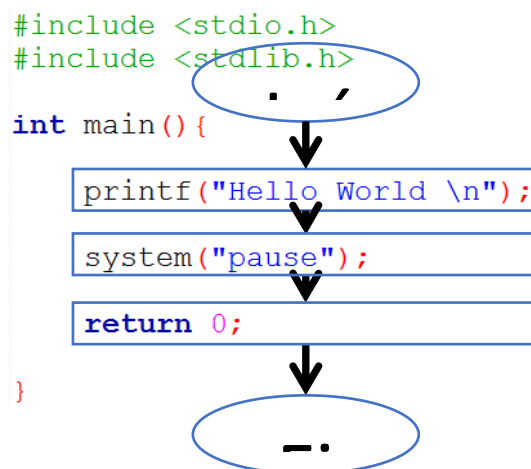
    printf("Hello World \n");

    system("pause");

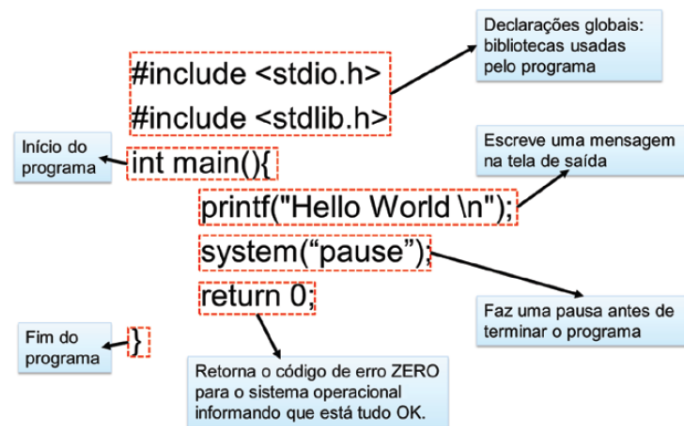
    return 0;

}
```

Fluxo de leitura e execução:



Conceito de cada elemento:



Como comentar seu código veja no exemplo abaixo como é simples:

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

int main() {
    /*
     * A função printf serve
     * para escrever na tela
     */
    printf("Hello World \n");

    //faz uma pausa no programa
    system("pause");

    return 0;
}
```

Estruturas de dados

Os dados na verdade são os valores fornecidos para resolução de um problema, esses valores podem ser inseridos pelo usuário, ou podem ser originados do processamento de cálculos, de arquivos, banco de dados e outros programas. Os dados são armazenados temporariamente em variáveis para que sejam processados de acordo com as especificações do algoritmo.

Tipos de Dados

Em programação é muito importante que se defina o tipo de dado que é mais adequado para ser armazenado em uma variável. É preciso que se tenha

conhecimento prévio da informação (DADO), para que se resolva o problema. Tendo essa premissa em mente, o algoritmo será desenvolvido de forma correta.

Para criar algoritmos usamos os tipos de dados primitivos (lógico, literal, inteiro e real), lembrando que algoritmos representam a resolução do problema. Já na criação de programas existem desdobramentos a fim de adequar os dados ao propósito de cada linguagem e a resolução prática dos problemas.

Na tabela a seguir vamos analisar algumas definições de tipos de dados primitivos e seus desdobramentos na linguagem de programação C. Nesta tabela estarão algumas definições, especificações de dados e observações importantes a respeito das particularidades da linguagem C, objetivo de ensino e aprendizagem deste tópico em especial.

Primitivos		Específicos para linguagem de programação C	
Tipos de dados	Definição	Tipos de dados	Capacidade de armazenamento na memória do computador, de acordo com a linguagem C
Literal – Também conhecido como texto ou caractere	Poderá receber letras, números e símbolos. <i>OBS: Os números armazenados em uma variável cujos tipos de dado Literal não poderão ser utilizados para cálculos</i>	<code>char</code>	16 bits – Armazena Unicode <i>OBS: Também é possível armazenar dados do tipo Literal na classe String.</i>
Inteiro	Poderá receber números inteiros negativos e positivos	<code>byte</code>	8 bits – De (–128) até (127)
		<code>short</code>	16 bits – De (–32.768) até (32.767)
		<code>int</code>	32 bits – De (–2.147.483.648) até (2.147.483.647)
		<code>long</code>	64 bits – De (–9.223.372.036.854.775.808) até (–9.223.372.036.854.775.807)
	Poderá receber números reais, isto é, com casas	<code>Float</code>	32 bits – De (–3, 4E–38) até (–3, 4E+38)

Real - também conhecido como flutuante	decimais, positivos ou negativos	<code>double</code>	64 bits – De $(-1, 7E-308)$ até $(+1,7E+308)$
Lógico – Também conhecido como booleano	Poderá receber verdadeiro (1) e (0) falso	<code>boolean</code>	8 bits – Em Java, pode-se armazenar <code>true</code> ou <code>false</code>

Variáveis

Quando um algoritmo é transcrito as variáveis são utilizadas para representar valores desconhecidos para resolução de problemas, esses valores poderão ser alterados de acordo com a situação. As variáveis também têm a função de armazenar os dados temporariamente na memória RAM.

As variáveis detêm as seguinte características:

- Posição de memória que armazena uma informação
- Pode ser modificada pelo programa
- Deve ser definida antes de ser usada

Características de identificação de variáveis para algoritmos:

- Não usar espaços entre as letras, ao invés disso utilizar underline (sublinhado): `nome_do_cliente` ou `nomecliente`.
- Não iniciar o nome da variável com números.
- Sem caracteres especiais (? /:@#ETC., Ç).
- Ser sucinto e usar coerência.
- Não utilizar palavras que são utilizadas para representar ações específicas dos algoritmos.

Características de identificação de linguagem C:

- Os nomes para variáveis em C se diferenciam em, NomeCliente, que é diferente de nomecliente e de nomeCliente.
- Nomes devem começar com uma letra, um caractere sublinha ou underline(_) ou (\$).
- Sem caracteres especiais (? /:@#ETC., Ç).
- Letras podem ser maiúsculas e minúsculas.
- Não podem usar palavras reservadas, float, for, int etc.
- Precisamos informar ao programa quais dados queremos armazenar
- Precisamos também informar o que são esses dados (qual o tipo de dado)
- Um nome de uma pessoa
- Uma cadeia de caracteres ("André" - 5 caracteres)
- O valor da temperatura atual
- Um valor numérico (com casas decimais)
- A quantidade de alunos em uma sala de aula
- Um valor numérico (número inteiro positivo ou zero)
- Se um assento de uma aeronave está ocupado
- Um valor lógico (ocupado: verdadeiro / desocupado: falso)

Nomes de variáveis Corretos:

- Contador, contador1, Teste_123, _teste, int1

Nomes de variáveis Errados.

- comp!, .var, int, 1contador, -x, Teste-123, x&

Lista de palavras chaves

São palavras reservadas para o uso da linguagem.

auto	break	case	char	const	continue	do	double
else	for	int	union	static	default	void	return
enum	goto	long	unsigned	struct	extern	while	sizeof
float	if	short	volatile	switch	register	typeof	

Tipos básicos em C

char:

Um byte que armazena o código de um caractere do conjunto de caracteres local
caracteres sempre ficam entre ‘aspas simples’!

```
char sexo; // pode receber 'M' ou 'F'
char UnidadeTemperatura; //pode receber 'C' para Celsius
                        //ou 'F' para Fahrenheit
char opcoes; // pode ser '1', '2' , '3' ou '4'
```

int:

um inteiro cujo tamanho depende do processador, tipicamente 16 ou 32 bits

```
int NumeroAlunos;
int Idade;
int NumeroContaCorrente;
int N = 10; // o variável N recebe o valor 10
```

Números reais

Tipos: float, double e long double

OBS: A parte decimal usa ponto e não vírgula!

float: um número real com precisão simples

```
float Temperatura; // por exemplo, 23.30
float MediaNotas; // por exemplo, 7.98
float TempoTotal; // por exemplo, 0.0000000032 (s)
```

double:

um número real com precisão dupla

Números muito grandes ou muito pequenos

```
double DistanciaGalaxias; // número muito grande
double MassaMolecular; // em Kg, número muito pequeno
double BalancoEmpresa; // valores financeiros
```

Tabela com os tipos

Tipo	Bits	Intervalo de valores
char	8	-128 A 127
unsigned char	8	0 A 255
signed char	8	-128 A 127
int	32	-2.147.483.648 A 2.147.483.647
unsigned int	32	0 A 4.294.967.295
signed int	32	-32.768 A 32.767
short int	16	-32.768 A 32.767
unsigned short int	16	0 A 65.535
signed short int	16	-32.768 A 32.767
long int	32	-2.147.483.648 A 2.147.483.647
unsigned long int	32	0 A 4.294.967.295
signed long int	32	-2.147.483.648 A 2.147.483.647
float	32	1,175494E-038 A 3,402823E+038
double	64	2,225074E-308 A 1,797693E+308
long double	96	3,4E-4932 A 3,4E+4932

Atribuição

Operador de Atribuição: =

O nome da variável recebe a expressão, valor ou contante

nome_da_variável = expressão, valor ou constante;



O operador de atribuição “=” armazena o valor ou resultado de uma expressão contida à sua **direita** na variável especificada à sua **esquerda**.

Ex.:

```
int main( ){
    int x = 5; // x recebe 5
    int y;
    y = x + 3; // y recebe x mais 3

    return 0;
}
```

A linguagem C suporta múltiplas atribuições

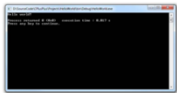
- `x = y = z = 0;`

Comando de saída

`printf()`

Significa:

- print formatted
- Comando que realiza a impressão dos dados do programa na tela



← `printf("texto");`

- O texto a ser escrito deve ser sempre definido entre “**aspas duplas**”

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

```
int main(){
    printf("Esse texto sera escrito na tela");

    return 0;
}
```

Especificadores de formato do printf()

Alguns tipos de saída	
%c	escrita de um caractere (char)
%d ou %i	escrita de números inteiros (int ou char)
%u	escrita de números inteiros sem sinal (unsigned)
%f	escrita de número reais (float ou double)
%s	escrita de vários caracteres
%p	escrita de um endereço de memória
%e ou %E	escrita em notação científica

Teste o exemplo:

```
int main() {
    printf("Esse texto sera escrito na tela");

    int x = 10;
    float y = 20;
    printf("%d", x);

    printf("%d %f", x, y);

    printf("Valor de x eh %d e o de y eh %f", x, y);

    return 0;
}
```