INTRODUÇÃO

A linguagem de Programação C é uma linguagem estruturada e criada em 1972, por Dennis Ritchie, no AT&T Bell Labs, para desenvolver o sistema operacional Unix.

Segundo Index Tiobe (2015) que ranqueia as linguagens de programação mais utilizadas no mundo, a Linguagem C é uma das mais populares e utilizadas, seguidas pela Linguagem Java e C++, conforme demonstrado na Tabela 1.

Tabela 1

As dez mais utilizadas Index Tiobe 2015

Feb 2015	Feb 2014	Change	Programming Language	Ratings	Change
1	1		C	16.488%	-1.85%
2	2		Java	15.345%	-1.97%
3	4	↑	C++	6.612%	-0.28%
4	3	↓	Objective-C	6.024%	-5.32%
5	5		C#	5.738%	-0.71%
6	9	↑	JavaScript	3.514%	+1.58%
7	6	↓	PHP	3.170%	-1.05%
8	8		Python	2.882%	+0.72%
9	10	↑	Visual Basic .NET	2.026%	+0.23%
10	-	⇈	Visual Basic	1.718%	+1.72%

Fonte: extraído de http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

A Linguagem C é uma linguagem compilada para o sistema operacional. Existem diferentes compiladores que executam esta função, entre os mais comuns estão:

- TDM-GCC (usamos este por ser do projeto GNU ou seja, software livre)
- LCC *Compiler*
- Intel C/C++ *Compiler*

A compilação é a tradução do código de alto nível escrito pelo programador em linguagem de baixo nível que o computador pode entender.

SINTAXE DA LINGUAGEM C

BIBLIOTECAS

As bibliotecas são necessárias para utilizarmos determinadas funções próprias da Linguagem C.

Devem ser incluídas no cabeçalho.

O comando para inserir bibliotecas é o #include.

FUNÇÃO MAIN

A função *MAIN* é a porta de entrada de qualquer aplicação, seja em linguagem C, Java, entre outras. A sintaxe pode variar entre as diferentes linguagens, porém é imprescindível a utilização da função *main* para iniciar a aplicação.

```
Int main () {
}
```

Todo código executável em C deve estar contido no método *main*, exceto bibliotecas e variáveis declaradas.

FUNÇÃO PRINTF

Esta função tem por objetivo exibir textos, caracteres, números, gráficos no console.

printf (“Bem vindo”);

Frases devem estar entre aspas duplas (string).

A função printf requer a utilização da biblioteca <stdio.h>

VARIÁVEIS

Variáveis são espaços abertos na memória do computador para armazenar dados. Podemos armazenar tipos de dados diferentes. Para declarar variáveis é simples, basta indicar o tipo de dado que será armazenado e um nome para a variável. Os principais tipos são:

- **int**

  O tipo de dado **int** (inteiro) serve para armazenar valores numéricos inteiros. Ocupa 32 bits. Escala: -2.147.483.648 a 2.147.483.647 (ambiente de 32 bits)

- **float**

  O tipo de dado **float** serve para armazenar números de ponto flutuante, ou seja, com casas decimais. Ocupa 32 bits. Escala: $3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$

- **double**

  O tipo de dado **double** serve para armazenar números de ponto flutuante de dupla precisão, tem o dobro do tamanho do float e portanto o dobro da capacidade. Ocupa 64 bits. Escala: $1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$

- **char**

  O tipo **char** serve para armazenar caracteres. Com vetores do tipo char é possível criar cadeias de caracteres (strings). Para a atribuição de valores em variáveis do tipo char, deve se colocar o dado entre aspas simples.

  Ocupa 8 bits. Escala: 128 a 127

- **void**

  O tipo **void** armazena vazios. Vazio é diferente de zero. Ocupa 0 bits. Escala: nenhum valor.

Exemplo:

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

int  numero = 1;
float  numeroFloat = 2.0;
double  numeroDouble = 2.0112;
char  letra = 'a';

int main() {

        printf(" o numero e: %d\n", numero);
```

```
        printf(" o numero e: %f\n", numeroFloat);
        printf(" o numero e: %f\n", numeroDouble);
        printf(" a letra e: %c\n", letra);
        system("pause");
}
```

Com exceção do void todos os tipos básicos de dados podem ser acompanhados por um modificador: short (curto), long (longo) e unsigned (apenas números positivos).

Exemplos: long double, short float, unsigned int, etc.

Nota: para a exibição de valores no console, utilizamos **%** no campo em que esta informação deve ser demonstrada. O **%** deve ser acompanhado da indicação do tipo do dado que será exibido, conforme abaixo:

%d – para inteiros

%f – para flutuantes (float e double)

%c – para caracter

%s – para string (cadeia de caracteres)

FUNÇÃO SCANF

A função scanf tem por objetivo a captura de dados digitados pelos usuários. Utiliza a biblioteca <stdio.h>

Ex:

```
int valor;

scanf("%d", &valor);

printf("O valor é: %d", valor);
```

O símbolo **&** ao lado da variável indica que queremos armazenar o valor na mesma.

Deste modo, podemos solicitar valores para o usuário digitar, armazenar estes valores em variáveis e efetuarmos operações aritméticas com as variáveis, exibindo um resultado no console, por exemplo:

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

float valor1, valor2, soma;
```

```
int main() {

	printf("Digite o primeiro valor: \n");
	scanf("%f", &valor1);
	printf("Digite o segundo valor: \n");
	scanf("%f", &valor2);
	soma = valor1 + valor2;
	printf("O valor da soma e: %0.2f\n", soma);
	system("pause");

}
```

As operações aritméticas suportadas pela linguagem são: + (soma), - (subtração), * (multiplicação), / (divisão) e % (resto de divisão)

Referências:

INDEX TIOBE. Disponível em: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html, acessado em 25/02/15

MIZRAHI, Victorine Viviane. Treinamento em Linguagem C, 2ª edição. São Paulo: Pearson Prentice Hall, 2008.

Prof. Emerson