Um programa é uma sequência de código organizada de tal forma que permita resolver um determinado problema. Um programa pode ser desenvolvido em módulos distintos e/ou em subprogramas. Dessa forma terá que existir um critério ou formato de escrita de programas bem definido que indique ao compilador, entre todo o conjunto de código escrito pelo programador, qual a instrução ou local onde irá começar a execução do mesmo.

A título de exemplo, um programa em PASCAL começa a execução na primeira instrução existente no único bloco BEGIN...END que não esteja associado quer a uma função quer a um procedimento. É absolutamente necessário que esse bloco BEGIN...END sejá único e se encontre depois de todas as funções e procedimentos.

No caso de C, existe uma função em que são colocadas todas as instruções que queremos que sejam executadas. Essa função chama-se main(), e todo o código a executar é colocado entre {}. Ao conjunto de código existente entre chaves chama-se Bloco.

Vamos então escrever o nosso primeiro programa em C.

Main(){

}

Nota: o nº de cada uma das linhas serve para facilitar a localização de código. Assim, deve-se apenas escrever o código respectivo.

Em seguida, podemos criar o arquivo executável correspondente e mandá-lo executar

Como pode se observar, este programa faz uma das coisas que mais gosto na vida, e que é – não fazer nada.

Observemos agora com maior atenção cada uma das linhas .

A primeira linha é composta pela palavra main, que é o local onde começam todos os programas em C.

Para indicar que se trata de uma função, a palavra main é seguida com parênteses\_main()\_, pois em C qualquer função tem que ser seguida por parênteses.

Os parênteses sem mais nada após o nome da função indicam que a função não recebe qualquer informação do mundo exterior.

É necessário não esquecer que C é Case Sensitive, isto é, faz diferenciação entre maiúsculas e minúsculas, não sendo portanto a mesma coisa escrever main(), Main(), MAIN() ou mAIN(). Todas as instruções de C são escritas com letra minúscula, e só se deve utilizar letras maiúsculas quando desejamos utilizar variáveis, mensagens ou funções escritas por nós.

Capítulo UM

O código executado pela função main() é escrito entre as chaves {}. No caso do nosso, programa, como não escrevemos qualquer instrução o programa simplesmente começa e acaba logo em seguida. E assim escrevemos o primeiro programa em C.

Nota: Caso seu compilador tenha apresentado um WARNING com uma mensagem semelhante a “Function should return a value” ou “main: no return value” é porque, em princípio, está-se usando um compilador de C++. Pode-se ignorar o Warning sem qualquer problema. Mas se o leitor quiser que desapareça coloque, antes da função main, a palavra void(que significa “nada”).

Nesse caso, o seu programa deverá ser escrito da seguinte forma:

void main(){

}

mais tarde você saberá o porquê.

É habitual que o primeiro programa escrito em qualquer linguagem seja para fazer a apresentação da linguagem ao mundo, Vamos então escrever o mais conhecido dos programas de C.

#include <stdio.h>

main(){

printf(“Hello Word”);

}

Depois de executado, esse programa proporciona a seguinte saída:

Hello World

Esse programa é um tudo semelhante ao anterior , com a exceção da existência de uma linha de código entre as chaves.

A linha 4: é a responsável pela apresentação da mensagem que queremos imprimir.

A mensagem que queremos imprimir é Hello World.

Sempre que queremos tratar conjuntos de caracteres temos que colocá-los entre aspas, para que sejam considerados como um todo “Hello World”.

Sendo C uma linguagem com muito poucas palavras reservadas, não é de surpreender que C não possua mecanismos de Entrada e Saída incorporados. Em vez disso, ela recorre à sua potente biblioteca de funções para fornecer esse tipo de serviço.

Uma das funções que permite a escrita na tela é a função **printf = print + f**ormatado. Tratando-se de uma função, tem que ser necessariamente escrita com parênteses – **printf()**.Dentro dos parênteses é feita a comunicação com a função. Nesse caso passamos a string (conjunto de caracteres) que queremos que seja escrita – **printf(“Hello World”)**;

Em C, cada instrução deve ser terminada com um ponto-e-vírgula(;), obtendo-se assim a linha nº4 do programa **printf(“Hello World”);**

Notar que o caractere aspas “ é um único caractere e não pode ser substituído pelo caractere aspas simples ‘ou por duas aspas simples’’

Como foi referido anteriormente, C não possui mecanismo de Entrada e Saída. Para resolver esse problema remos que recorrer a um conjunto de funções que existem em **Bibliotecas de Funções.** Isso implica que temos que adicionar à linguagem um conjunto de outras funcionalidades que, por defeito, ela não nos proporciona.

Para ter acesso a esse conjunto de funções teremos que incluir a sua definição no nosso programa.

A linha **#include<stdio.h>**  não é C, mas sim uma diretiva que indica o compilador (mais propriamente ao pré-processador) que deverá adicionar ao processo de compilação um arquivo existente em alguma parte no disco do seu computador, chamado **stdio.h** de forma que o compilador tenha acesso a um conjunto de informações sobre funções que virá a utilizar.

Esses arquivos têm sempre a extensão .h, pois não têm código, mas apenas os cabeçalhos(headers) das funções que representam. São por isso habitualmente designados por header files.

Desse modo, a linha **#include<stdio.h>**  significa “adiciona o arquivo **stdio.h** ao meu programa”. Exatamente nessa posição. Como não é uma instrução de C, não é seguida de;

O arquivo stdio.h permite o acesso a todas as funções de entrada e saída normais; **stdio** quer dizer **standard input/output.**