机器指令：计算机能直接识别和接受的二进制代码。

机器语言：机器指令的集合。

汇编程序软件的作用:把符号语言的指令转换为机器指令，一般，一条符号语言的指令对应一条机器指令。

计算机低级语言：指依赖于具体机器特性，面向机器的语言，例：机器语言和汇编语言。

计算机高级语言：语言能力很强，且不依赖于具体机器，用它写出的程序对任何型号的计算机都适用，与具体机器距离较远的语言。

计算机不能直接识别高级语言程序，需要进行“翻译”，用一种编译程序的软件把高级语言写的程序（源程序）转换为机器指令的程序（目标程序），然后让计算机执行机器指令程序，最后得到结果。高级语言的一个语句往往对应多条机器指令。

C语言程序的结构

1. 一个程序由一个或多个源程序文件组成。在一个源程序文件中可以包含三个部分。

1.预处理指令。如#include <stdio.h>，就是将stdio.h头文件的内容读进来，放在#include指令行，取代#include <stdio.h>。

2.全局声明：即在函数之外进行的数据声明。全局变量：在函数外面声明的变量称为全局变量。

在程序开头（定义函数之前）声明的变量，则在整个源程序文件范围内有效。在函数中声明的变量是局部变量，只在函数范围内有效。

3.函数定义。

（2）函数时c程序的主要组成部分。一个c语言程序是由一个或多个函数组成的，其中必须包含一个main函数（且只能有一个）

（3）一个函数包括两个部分。

1.函数首部，即函数的第一行。

2.函数体，即函数首部下面的花括号内的部分，如果在一个函数中包含有多层花括号，则最外面的一对花括号是函数体的范围。分为声明部分和执行部分。

声明部分：定义在本函数中所用到的变量；对本函数所调用的函数进行声明。

执行部分：由若干个语句组成，指定在函数中所进行的操作。

如：void dump()

{}

它是一个空函数，什么也不做，既无声明部分也无执行部分，但这是合法的。

（4） 程序总是从main函数开始执行的。（不论main函数在整个程序中的位置如何）

（5）程序中对计算机的操作是由函数中的c语句完成的。C程序书写格式是比较自由的，一行内可以写几个语句，或把一个语句分写在多行上，但为清晰起见，习惯上每行只写一个语句。

（6）在每个数据声明和语句的最后必须有一个分号。

（7）C语言本身不提供输入输出语句。