C 语言补充知识

（本文档默认读者至少学过 Java 或 Python 中的一种语言）

C 语言诞生于上个世纪 70 年代，其设计目标是提供一种能以简单方式编译、处理低级存储器并保持高的运行效率的编程语言。C++是在 C 的基础上增加了“类（class）”等面向对象的功能而形成的更高级语言，因此 C 与 C++的语法规则基本是相同的，所以我们经常可以看到一些程序员的招聘要求会写“熟练使用 C/C++语言”。尽管当下工业界工程开发中使用 JAVA 语言最多，但因 C/C++编写出来的程序运行效率很高（运行时间很快、占用资源很少），仍然在很多对效率要求高的场景下广泛使用，例如嵌入式设备、操作系统和驱动程序等。

1. C 程序结构

一个 C 程序主要包括以下部分：

- 预处理器指令
- 函数
- 变量
- 语句 & 表达式
- 注释

下面以一个简单的输出“Hello World”的程序为例

```c
#include <stdio.h>
/* 上面这一行是预处理器指令，功能可以理解为Python或JAVA中的import某个库 */

int main()/* main函数为C程序开始执行的地方，不像Python编译器可以自动找到程序该从哪里执行 */
{         /* 两个花括号内的部分为main函数的函数体，Python采用缩进取代了C的花括号 */

   printf("Hello, World! \n");/*输出Hello World*/

   return 0;  /*返回0*/
}
```

2. 数据类型

本文档开篇部分有提到 C 语言的运行效率高，运行效率高的一个表现便是运行时消耗的资源少，这里的资源便包括了存储空间。在 Python 中我们可以随意定义一个变量，而不用考虑它占用多少存储空间。但在 C 语言中，我们定义一个变量的同时必须说明给它分配多少存储空间，这种向编译器说明分配多少存储空间的方式便是在定义变量时指定它的数据类型。数据类型除了可以表示分配给变量多少存储空间外，还告诉了编译器这个变量在存储空间的每一个 0 或 1 比特所代表的意义。C 语言中有很多种数据类型，我们自己还可以根据需要在程序中定义新的数据类型，这一节只介绍我们课程涉及的 C 语言中常用的一些基本数据类型。

整数类型：存储的变量为整数，例如 0,1,2，-1，-2

类型名称	占字节数	其他叫法	表示的数据范围
char	1	signed char	-128 ~ 127
unsigned char	1	none	0 ~ 255
int	4	signed int	-2,147,483,648 ~ 2,147,483,647
unsigned int	4	unsigned	0 ~ 4,294,967,295
short	2	short int	-32,768 ~ 32,767
unsigned short	2	unsigned short int	0 ~ 65,535
long	4	long int	-2,147,483,648 ~ 2,147,483,647
unsigned long	4	unsigned long	0 ~ 4,294,967,295

浮点类型：存储的变量包含小数部分，例如 1.1,3.75，-4.9,2.0

类型	存储大小	值范围	精度
float	4 字节	1.2E-38 到 3.4E+38	6 位小数
double	8 字节	2.3E-308 到 1.7E+308	15 位小数
long double	16 字节	3.4E-4932 到 1.1E+4932	19 位小数

3. C 函数

C 语言中的函数定义的一般形式如下:

```
return_type function_name( parameter list )
{
   body of the function
}
```

下面以一个简单的求两数相加之和的函数为例:

```
1   int two_plus(int num1 ,int num2)
2   {
3       int num3;
4       num3=num1+num2;
5       return num3;
6   }
```

第 1 行的第 1 个 int 为函数的返回值类型，即 return_type，指的是函数体中第 5 行 return 后面紧跟的那个变量的数据类型。

当然，return 后面也可以不跟任何变量，这时候因为函数没有返回值，故返回值值类型为空，我们用 void 来表示返回值为空的情况，例如下面这个程序:

```
1   void main()
2   {
3       printf("Hello World");
4       return;
5   }
```

4. 指针

在第 2 节中我们介绍了 C 语言中的数据类型，指针也是 C 语言的一种数据类型，因此指针类型的变量也会有自己的存储空间，而在指针类型的变量存储空间中存储的 0 或 1 比特的意义是指针所指向的变量的存储地址。

下面以一段代码来解释这个概念:

```c
#include <stdio.h>

int main ()
{
   int  var = 20;   /* 实际变量的声明 */
   int  *ip;        /* 指针变量的声明 */

   ip = &var;  /* 在指针变量中存储 var 的地址 */

   printf("Address of var variable: %p\n", &var  );

   /* 在指针变量中存储的地址 */
   printf("Address stored in ip variable: %p\n", ip );

   /* 使用指针访问值 */
   printf("Value of *ip variable: %d\n", *ip );

   return 0;
}
```

在第 5 行，我们定义了一个类型为 int 的变量 var

在第 6 行，我们定义了一个指针变量 ip，它的数据类型为"int*"，表面它可以存储类型为 int 的变量的地址。假如我们希望 ip 存储类型为 char 的变量的地址，我们可以把第 6 行改为"char *ip;"，以此类推

在第 8 行，我们先用"&var"取得了变量 var 的地址，然后把这个地址赋值给了 ip，这时候我们称 ip 指向了 var。

在第 16 行，我们使用*ip 获取 ip 指向的地址中所存储的值，也就是 var 的值

上面的程序运行最终结果为:

```
Address of var variable: bffd8b3c
Address stored in ip variable: bffd8b3c
Value of *ip variable: 20
```

5. 占位符

在上面的程序的第 10/13/16 行的 printf（"…"）中，出现了%p 和%d，这是 C 程序中的占位符。顾名思义，占位符就是先占住一个固定的位置，等着你再往里

面添加内容的符号。不同的数据类型使用的占位符也不同，它的的用法如下：

- %d, %i 代表整数
- %f 浮点
- %s 字符串
- %c char
- %p 指针
- %fL 长log
- %e 科学计数
- %g 小数或科学计数。
- %a,%A 读入一个浮点值(仅C99有效)。
- %c 读入一个字符。
- %d 读入十进制整数。
- %i 读入十进制，八进制，十六进制整数。
- %o 读入八进制整数。
- %x,%X 读入十六进制整数。
- %s 读入一个字符串，遇空格、制表符或换行符结束。
- %f,%F,%e,%E,%g,%G 用来输入实数，可以用小数形式或指数形式输入。
- %p 读入一个指针。
- %u 读入一个无符号十进制整数。
- %n 至此已读入值的等价字符数。
- %[] 扫描字符集合。
- %% 读 % 符号