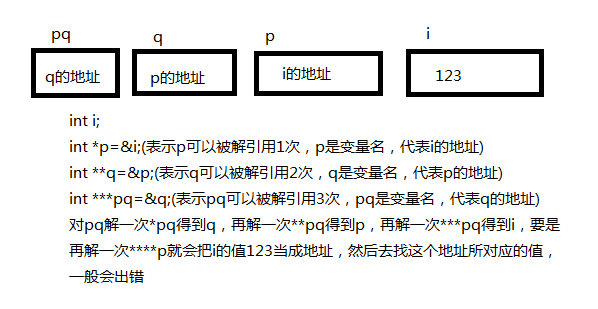
1、int \*\*\*pp = &ip; （pp表示ip这个变量的地址（pp==&ip），\*\*\*表示pp可以被解引用3次，解一次引用表示pp这个地址变量所指向的值）



**指向指针的指针**

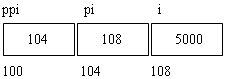
在指针初始化的第5种方式中提到了用一个指针的地址来初始化一个指针。回忆一下上一讲的内容：指针是一种变量，它也有自己的地址，所以它本身也是可用指针指向的对象。我们可以将指针的地址存放在另一个指针中，如：

int i = 5000;

int \*pi = &i;

int \*\*ppi =&pi;

此时的ppi即是一个指向指针的指针,下图表示了这些对象：



i的地址为108，pi的内容就是i的地址，而pi的地址为104，ppi的内容即是pi的地址。对ppi解引用照常会得到ppi所指的对象，所获得的对象是指向int型变量的指针pi。想要真正地访问到i.，必须对ppi进行两次解引用，如下面代码所示：

printf("%d\n", i );

printf("%d\n", \*pi );

printf("%d\n", \*\*ppi );

以上三条语句的输出均为5000。

数组名是常指针，不能对它的值进行修改。ia + 99是可以的，但ia++是不行的，它的意思是ia = ia +1;修改了ia的值。

在想函数中传递指针或数组是要看是要对指针本身做改变还是对指针所指向内容做改变，若为前者，应该把指针的指针传进来；若为后者，直接传进来指针就能操作。C语言中永远都是值传递，就是把原来的值（包括指针）拷贝一份给形参，无论对你拷贝的这份值作何改变，都不会影响原来的值。