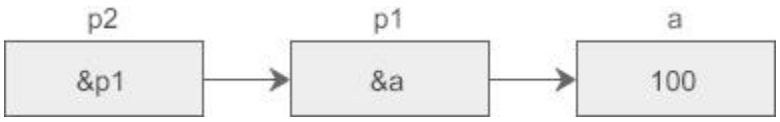
C 语言二级指针（指向指针的指针）

指针可以指向一份普通类型的数据，例如 int、double、char 等，也可以指向一份指针类型的数据，例如 int *、double *、char * 等。

如果一个指针指向的是另外一个指针，我们就称它为二级指针，或者指向指针的指针。

假设有一个 int 类型的变量 a，p1 是指向 a 的指针变量，p2 又是指向 p1 的指针变量，它们的关系如下图所示：



将这种关系转换为 C 语言代码：

```
int a =100;
int *p1 = &a;
int **p2 = &p1;
```

指针变量也是一种变量，也会占用存储空间，也可以使用&获取它的地址。C 语言不限制指针的级数，每增加一级指针，在定义指针变量时就得增加一个星号*。p1 是一级指针，指向普通类型的数据，定义时有一个*；p2 是二级指针，指向一级指针 p1，定义时有两个*。

如果我们希望再定义一个三级指针 p3，让它指向 p2，那么可以这样写：

```
int ***p3 = &p2;
```

四级指针也是类似的道理：

```
int ****p4 = &p3;
```

实际开发中会经常使用一级指针和二级指针，几乎用不到高级指针。

想要获取指针指向的数据时，一级指针加一个*，二级指针加两个*，三级指针加三个*，以此类推，请看代码：

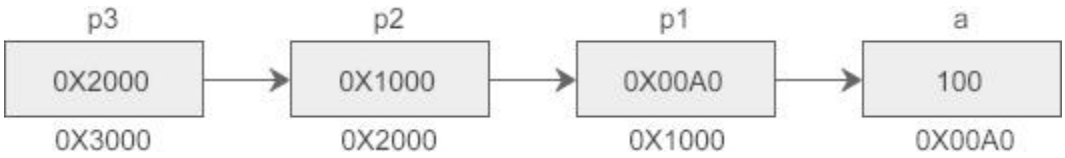
```
#include <stdio.h>

int main(){
    int a =100;
    int *p1 = &a;
    int **p2 = &p1;
    int ***p3 = &p2;

    printf("%d, %d, %d, %d\n", a, *p1, **p2, ***p3);
    printf("&p2 = %#X, p3 = %#X\n", &p2, p3);
    printf("&p1 = %#X, p2 = %#X, *p3 = %#X\n", &p1, p2, *p3);
    printf(" &a = %#X, p1 = %#X, *p2 = %#X, **p3 = %#X\n", &a, p1, *p2, **p3);
    return 0;
}
```

运行结果：

```
100, 100, 100, 100
&p2 = 0X28FF3C, p3 = 0X28FF3C
&p1 = 0X28FF40, p2 = 0X28FF40, *p3 = 0X28FF40
 &a = 0X28FF44, p1 = 0X28FF44, *p2 = 0X28FF44, **p3 = 0X28FF44
```

以三级指针 p3 为例来分析上面的代码。***p3 等价于*(*(*p3))。*p3 得到的是 p2 的值，也即 p1 的地址；*(*p3) 得到的是 p1 的值，也即 a 的地址；经过三次“取值”操作后，*(*(*p3)) 得到的才是 a 的值。

假设 a、p1、p2、p3 的地址分别是 0X00A0、0X1000、0X2000、0X3000，它们之间的关系可以用下图来描述：



方框里面是变量本身的值，方框下面是变量的地址。