/\*

1. 指针就是地址，地址就是指针

2. 地址就是内存单元的编号

3. 指针变量 是存放地址的变量

4. 指针和指针变量是两个不同的概念，但是要注意： 通常我们叙述时，我们会把指针变量简化为指针，实际它们的含义并不一样。

指针的本质就是一个操作受限的非负整数。（只是一个编号，不能加乘除，可以相减）

\*/

#include <stdio.h>

int main(void)

{

int \* p;// p是变量的名字 ， int\* 表示p变量存放的是int 变量的地址

// int\*p; 不表示定义了一个名字叫做\*p的变量

// int\*p; 应该这样理解： p是变量名， p变量的数据类型是int\*类型

// 所谓int\*；类型就是存放int变量地址的类型

int i = 3;

p = &i;

/\*

1. p保存了i的地址， 因此p指向了i;

2. p 不是 i，i 也不是p, 更准确的说：修改p的值不影响i的值，修改i的值也不影响p的值

3. 如果一个变量指向了某个普通变量， 则\*指针变量 就完全等同于 普通变量

例子：

如果p是一个指针变量，并且p存放了普通变量i的地址

则p指向了普通变量i

\*p 就完全等同于i

或者说： 在所有出现\*p的地方都可以替换成i

在所有出现 i的地方都可以替换成\*p

\*p就是以p的内容为地址的变量

\*/

**指针**

1. **表示一些复杂的数据结构**

**2. 快速的传递数据**

**作用 3. 使函数返回一个以上的值、**

1. **能直接访问硬件**
2. **能够方便处理字符串**

**6. 是理解面向对象语言中引用的基础**

**总结：指针是C语言的灵魂**

**指针的定义：**

**地址： 内存单元的编号**

**从零开始的非负整数**

**范围：4G （32位）**

**从零开始的非负数**

**（0-4G (2的32次方字节)）**

**指针的基本类型：**

**\* 的含义**

1. **乘法**
2. **定义指针变量  
   int\* p //定义了一个名字叫p的变量，int\* 表示P 只能存放int 变量的地址。**
3. **指针运算符，取地址的逆运算符，取内容运算符  
   该运算符放在已经定义好的指针变量前面，如果p是一个已经定义好的指针变量  
   则\*p 表示以p的内容为地址的变量**

如何通过被调函数修改主调函数普通变量的值

1. 实参必须为该普通变量的地址
2. 形参必须为指针变量
3. 在被调函数中通过\*形参名=........  
   的方式就可以修改主调函数相关变量的值

**指针和数组**

**指针和一维数组**

**数组名**

**一维数组名是个指针常量， 它存放的是一维数组第一个元素地址。**

**下标和指针的关系**

**数组名是指针常量 数组名和第一个元素地址相铜他们是等价的，而下标p【i】就是\*（p+i）----如果p是个指针变量，则 p[i] 永远等价于\*（p+i）**

**指针变量的运算**

**指针和二维数组**