**指针与数组（不要加&这个符号哦）**

1. 指向数组元素的指针变量
2. 类型需要相等
3. 赋值
   * 1. Int a[] = {1, 2, 3};
     2. Int \*p = a;

3、

1. 指针引用数组元素
2. 回顾
   * 1. 记得我们上节所说的 \* & 符号吗
     2. \*p = 15； //相当于把a[0] 赋值为15
3. (p+i) 和 (a+i)
   * 1. p+i ==> 代表的指向下i个元素的地址
     2. a + i ==> 代表也是指向下i个元素
4. \*(p+i) 和 \*(a+i)
   * 1. 就是取值的问题
     2. \*(p+1) = 12; //就是a[1] = 12;
5. p[i] 和 a[i]
   * 1. 其实本质是一样的
6. 注意
   * 1. Int \*p = a； p++ 可以， 但是 a++ 就不行，不能让数组首地址进行++
7. 数组名作为函数参数
8. 实参用数组名， 形参用指针变量
9. 实参和形参都用指针变量

1. 指针与多维数组
2. 介绍
   * 1. Int a[3][4];
     2. a 代表第一行首地址
     3. a +1 代表第二行的首地址
     4. …………………………
     5. a[0] == &a[0][0] 第一行首地址
     6. a[1] == &a[1][0] 第二行首地址
3. 我们要指向1行2列 [现在我们要上升到二维的空间来了 （指向的都是地址）
   * 1. a[0][1] == a[0] + 1 == (\*a + 1 ) ==> a[0] = \*(a+0)
     2. a[1] == \*(a+1) == a[1][0]
     3. 那我们现在要取到 第二行第三列
        1. &a[1][2] == \*(a+1)+2 == a[1]+2
4. 代码

int a[3][4];

//\*a 代表的是a[3][4]的首地址

\*(\*a) = 10; //对a[0][0] 进行赋值操作

\*(\*(a+1)+3) = 80; //对a[1][3] 继续赋值操作

1. 指针数组和数组指针（我们按顺序来进行读就可以了）
2. 指针数组
   * 1. Int \*p[10]; //相当于定义了十个指针变量，每个都可以指向整型
     2. 代码

//指针数组

char \*p[4]; //代表了我们定义了4个指针变量

\*p = "b";

\*(p+1) = "s";

\*(p+2) = "s";

\*(p+3) = "x";

for(int i = 0; i < 4; i++){

printf("%s ", \*(p+i));

}

1. 数组指针
   * 1. int (\*p)[10]; // 数组指针，相当于一个指针变量.包含十个元素【一般说来，二维数组用这个比较多】
     2. 代码

//数组指针

int (\*p)[4];

p = a; //把a 的首地址赋值给p

for(int i = 0; i < 3; i++){

for(int j = 0; j < 4; j++){

printf("%d ", \*(\*(p+i)+j));

}

printf("\n");