引用（为了消除指针）

- **引用是一种声明关系，声明的时候必须要初始化，引用不开辟空间（不分配内存）**

```
int a = 0;

int& ra = a;   /* &ra是地址，ra是变量的值 */

int& rr = ra; /* 对引用的引用 */
```

- **此种声明关系，一经声明，不可变更**
- **可以对引用再次引用。多次引用的结果，多个引用指向用一个变量**
- **&前面有数据类型，则为引用。其他皆为取址符**

```
void swap (int *pa, int*pb)      /* 使用指针进行交换 */
{
    *pa ^= *pb;
    *pb ^= *pa;
    *pa ^= *pb;
}

void swap (int& ra, int& rb)     /* 使用引用进行交换 */
{
    ra ^= rb;
    rb ^= ra;
    ra ^= rb;
}
```

- **引用的本质，就是对指针的再次包装。指针是有引用的，但是，不可以对引用取地址（引用不能有指针）**

```
int *pp = "ming";

int* & rp = pp;
```

- **可以定义指针的引用，但不能定义引用的引用（**int && rra = ra; //**错误）**
- **可以定义指针数组，但不能定义引用数组，可以定义数组引用**

```
int x, y, z;

int *array[] = {&x, &y, &z};   //指针数组

int &array[] = {x, y, z};      //引用的数组（error）
```

```
int arr[] = {1, 2, 3, 4, 5}
int (&r_array)[5] = arr;        //数组引用
```

- **常引用 (** const & **)**

```
const int a = 0;

const int &ra = a;
```

-
-
-
-
-
-
-