```
//（2）智能指针总述
int *p = new int();
int *q = p;
int *r = q; //只有p,q,r都不再使用了的时候，才能释放掉这段内存。
//new /delete的写法要非常小心，防止早早的释放，也防止忘记释放，总之：用好并不容易；
//p 裸指针:直接用new返回的指针,这种指针，强大，灵活，但是开发者全程负责维护，一个不小心，就容易用错；一旦用错，造
//智能指针：解决裸指针可能代码的各种问题；
//智能指针，就理解成堆“裸指针”进行了包装，给裸指针外边包了一层；包装后为我们带来优点：
    //最突出的优点：智能指针 能够“自动释放所指向的对象内存”，大家再也不用担心自己new出来的内存忘记释放了；
//建议优先选择智能指针。使用智能指针的程序更容易编写和调试；
```

```
//c++标准库有四种智能指针：std::
    //a)auto_ptr(c++98)； unique_ptr(c++11)； shared_ptr(c++11) ；weak_ptr(c++11)；
//帮助我们进行动态分配对象（new出来的对象）的生命周期的管理。能够有效防止内存泄漏；
//目前auto_ptr已经完全被unique_ptr，所以大家不要再使用auto_ptr；c++11标准中反对使用auto_ptr（弃用）；
//这三种智能指针都是类模板，我们可以将new获得地址赋给他们；
//a）shared_ptr：共享式指针。多个指针指向同一个对象，最后一个指针被销毁时，这个对象会被释放。
    //weak_ptr是辅助shared_ptr工作的；
//b）unique_ptr：独占式指针；同一个时间内，只有一个指针能够指向该对象。
        //当然，该对象的所有权还是可以移交出去的；
//你忘记delete的时候，智能指针帮助你delete，或者说，你压根就不再需要自己delete；智能指针的本份（帮助你delete）；
```