

一、

- 1, 野指针：指针变量指向非法的内存空间
- 2, 可能导致的问题：
 - 1) 数据损坏：覆盖内存中的其他数据，导致数据损坏
 - 2) 程序崩溃：访问无效的内存地址，程序可能会崩溃
- 3, 避免野指针可以初始化指针，定义有效的内存地址，NULL 可能都会比野指针更好

二、

```
const int* p1;.....
```

相当于 int 定住：指针的指向&a &b 可以修改，但指针指向的值（地址）*p1 不能修改
int* const p2;

相当于 p2 定住：指针指向的值（地址）*p1 可以修改，但指针的指向&a &b 不能修改
const int* const p3;

int 与 p3 都定住

三、

```
void print_ptr(int* p)
{
    *p = *p + 1;
    cout << *p;
}
```

问题：没有检查*p 是否合法，如果不合法输出会报错！

改进：

```
void print_ptr(int* p)
{
    if (p == nullptr) {
        cout << "Error: null pointer passed to print_ptr";
        return;
    }

    *p = *p + 1;
    cout << *p;
}
```