```
//前面程序已判断处理了 Append，Insert，Write，"1,$n"以及指令为空的情况

int start, end;
char comma, type, res;
stringstream ss(cmd);       //将 cmd 转化为 stringstream 类型，便于通过流输入析取指令的不同部分

ss >> start;                //试图析取指令开头的数字至 start
if (ss.eof()) {             //若成功读取数字 start 并且指令已经结束
//（如果指令为空，在前面代码中已加入判断并抛出异常；如果指令不为空但未读取成功，则 badbit 置为 true 但未达到 eof）

    cmdNull(start);         //判断为 Null Command，调用对应函数
    return;                 //结束对本条指令的解析
}

ss >> comma >> end >> type;
//针对"?,?x"型指令,试图从指令中析取','至 comma、析取第二个数字至 end、析取操作类型至 type

if (ss.good() && comma == ',' && !(ss >> res)) {
//若上述两次析取均成功（goodbit 为 true），且指令满足"?,?x"形式(数字以逗号分隔，无多余字符）

    if (type == 'n') {                  //若指令以 n 结尾
        cmdNumber(start, end);          //判断为 Number Command，调用对应函数
        return;                         //结束对本条指令的解析
    } else if (type == 'd') {           //若指令以 d 结尾
        cmdDelete(start, end);          //判断为 Delete Command，调用对应函数
        return;                         //结束对本条指令的解析
    }
}
throw "Bad/Unknown command";            //不是规定的某个指令，抛出异常
```