面向对象程序设计基础作业十一 设计文档

1. 模型部分
   1. 功能简述

本程序能够判断字符串是否为合法整数（绝对值在1019以内的整数），用户可以多次输入待检验数据，输入END[[1]](#footnote-1)结束程序。设立上限主要是为了贴近long long类型的上限。关于“什么不是合法整数”的例子，可以参照下文异常说明部分。

* 1. 程序结构

本程序包含一个main.cpp文件作为执行的入口；一组string\_validator.cpp/.h，定义了类string\_validator，包含一个静态公有接口run()用于执行程序；一组exception.cpp/.h，定义了本程序所有可能使用的异常供抛出。

* 1. 异常说明

本部分列出了所有exception.h中定义了的异常。所有未在本部分说明的情形都视作合法整数。以下异常的排序代表了程序中抛出异常的优先级，越靠前的异常越优先抛出。如果一个输入同时触发了多个异常，仅有优先级最高的异常会被显示。

* + 1. EXC\_minus\_zero

这一异常用于表示输入是“-0”。我们认为，这不是一个合法整数：其标准表示法应该是“0”。

* + 1. EXC\_unexpected\_zero

这一异常用于表示输入的开头有若干连续的零，或输入的开头是负号、之后有若干连续的零。

* + 1. EXC\_bad\_char

这一异常用于表示输入中有数字、“-”（负号）、“.”（小数点）以外的符号，或者有大于一个小数点。

* + 1. EXC\_minus（与EXC\_bad\_char优先级相同[[2]](#footnote-2)）

这一异常用于表示输入中有不在开头的负号。

* + 1. EXC\_decimal

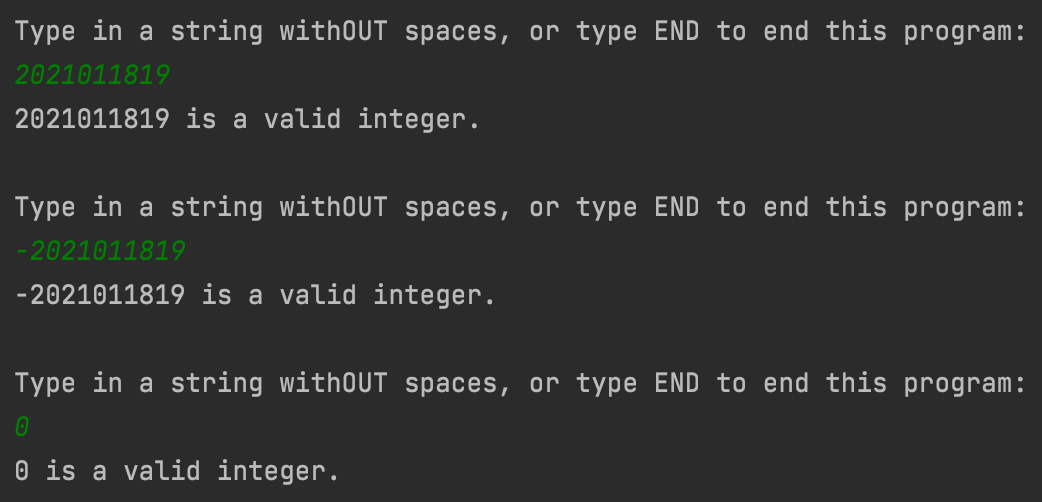
这一异常用于表示输入是一个合法的小数，但不是合法的整数。我们认为“123.00”不是一个合法整数：尽管它和合法整数123数值上相同，但它在有效数字上和123不同。另外，出于传统考虑，我们把形如.86这样省略整数部分0的小数视作一个合法小数（即报此错误而不报bad\_char），并把形如123.这样在整数结尾加点表示数字终结的数视作合法整数（在输出时会删除这一个小数点）。

* + 1. EXC\_overflow

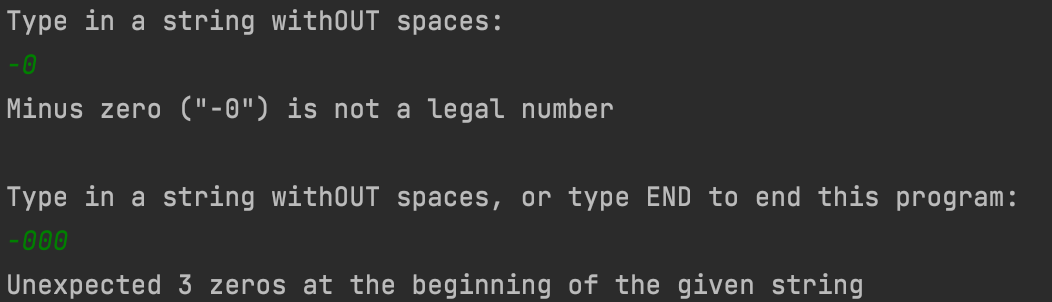
这一异常用于表示虽然输入是一个整数，但这个整数的绝对值太大（大于1019），所以在本程序中不合法。

1. 验证部分

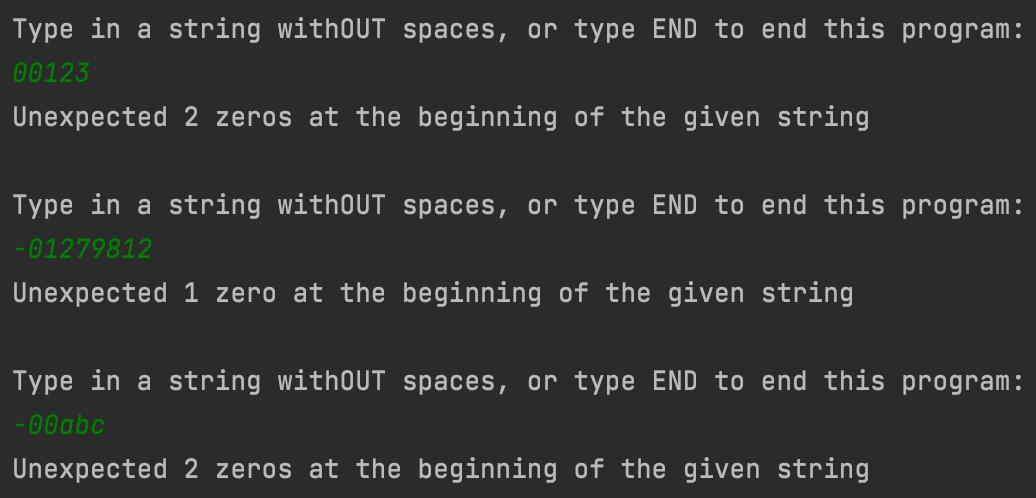
首先验证正常的输入输出：



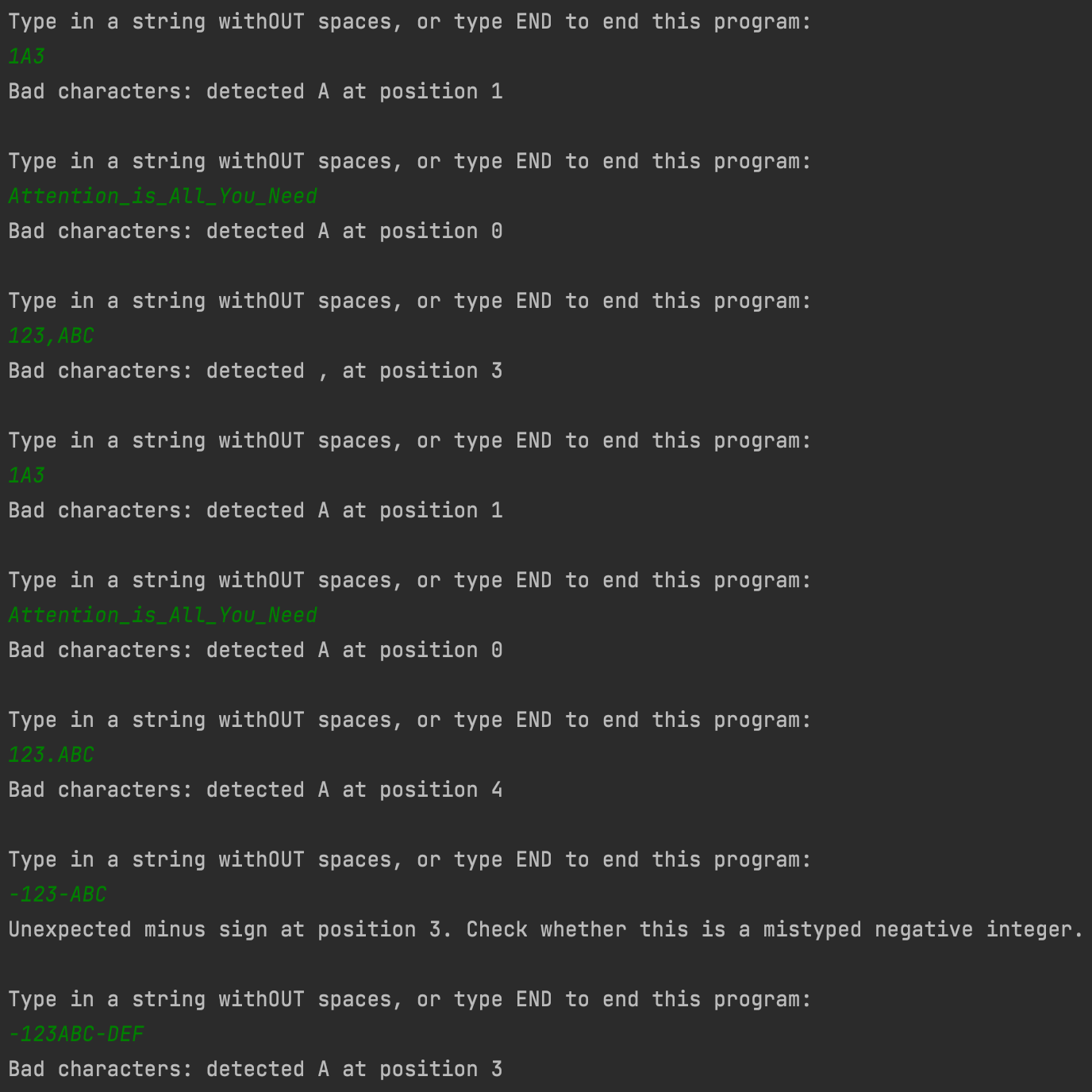
下面将验证各异常情况。



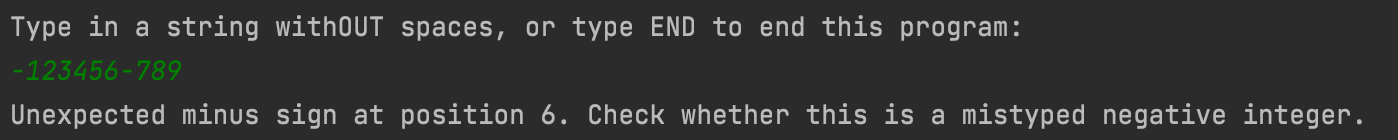
验证了-0会被正常抛出，但-000会被认为是连续零异常。



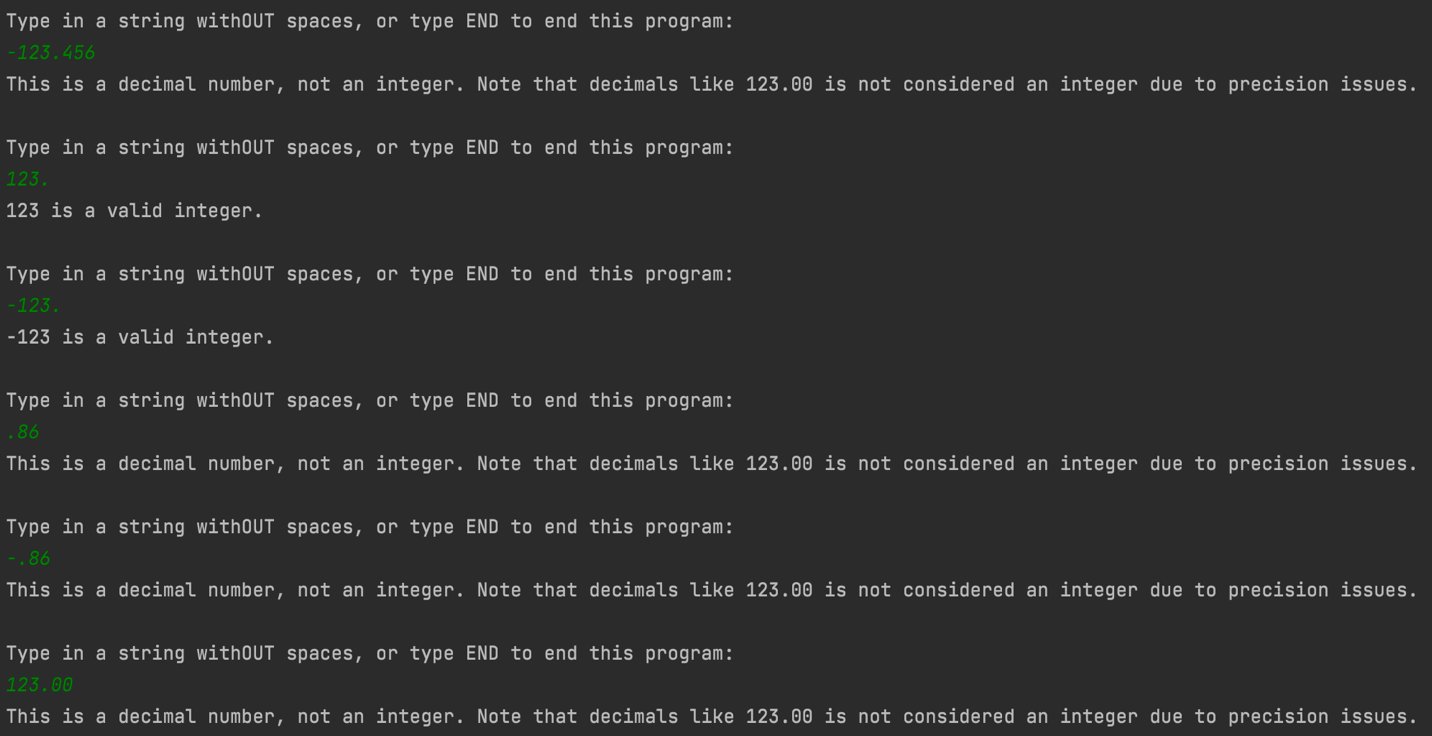
验证了连续零异常，且说明了这一异常比非法字符优先级更高。



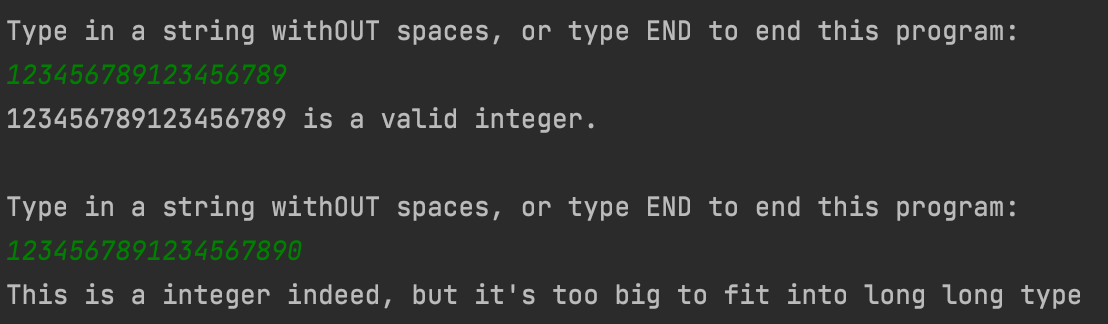
验证了非法字符异常，同时说明了非法字符异常和过多负号异常的优先级顺序，以及这两个异常和小数异常的优先级顺序。



验证了过多负号异常。

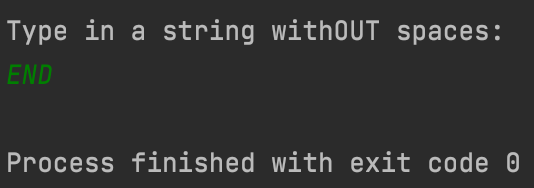


验证了小数异常，同时考虑了多个特殊情况。



验证了溢出异常，十八位数可以正常输出，但十九位数会被拒绝。

最后，验证程序的正常退出：



1. 为了保持程序的完整性，在此说明：END这一字符串本身不是一个合法整数。另外，本程序不能对含有空格的字符串进行处理，在此说明：所有含有空格的字符串不是一个合法整数。 [↑](#footnote-ref-1)
2. 具体报的异常取决于输入顺序，即若第一个非法字符更靠前则报EXC\_bad\_char，第二个负号更靠前则报EXC\_minus。 [↑](#footnote-ref-2)