

为了方便选手处理高精度小数，我们提供了一个定点高精度小数类。

该类可以对高精度小数进行加、减、乘、除、与 `double` 和整数类型互转、与字符串类型互转、比较等操作。

该类定义了一个值 P ，表示每次计算的绝对误差不超过 10^{-P} 。

任何一种该类的操作的时间复杂度都不超过 $O(P)$ 。

任何一个该类的实例使用的空间都不超过 $O(P)$ ，准确的说，一个实例最多会占用 $\frac{4P}{9} + 16$ 个字节的空间。

需要注意的是，高精度小数不能与 `double` 类型进行乘、除运算。

在调用任何一个跟该类有关的函数时，需要满足以下条件，否则不保证该类能正常工作。

- 1、 在任何计算过程中，用该类表示的数的绝对值不能超过 10^{18} ；
- 2、 `int/longint` 类型的参数的绝对值不能超过 10^9 ；
- 3、 `long long/int64` 类型的参数的绝对值不能超过 10^{18} ；
- 4、 `double` 类型的参数必须为有效的实数，且绝对值不能超过 10^9 ；
- 5、 字符串类型的参数必须为一个合法的实数，即，先是符号部分（负数为负号，正数没有），然后是一个纯数字串表示整数部分，然后是小数点和一个纯数字串表示小数部分，小数点和小数部分可以同时省略，整个实数的绝对值不能超过 10^9 ；
- 6、 除法的除数不能为 0 ，转字符串时的参数(小数位数)必须大于 0 。

在本题目录下，*drink_sample.cpp/c/pas* 是一个已经包含了该类的空程序，你可以在这个程序的基础上进行修改，也可以不使用这个程序。

在本题目录下，*decimal_test.cpp/c/pas* 是一个使用了该类的示例程序，你可以通过这个程序来了解该类的具体接口和实现细节，以及每个功能的使用方法。

对 C++ 语言的选手：

drink_sample.cpp 的第 9 行，“`const int PREC = 2100;`”，定义了 P 。

对 C 语言的选手：

drink_sample.c 的第 9 行，“`#define PREC 2100`”，定义了 P 。

在调用 `to_string_d(Decimal, char*, int)` 将高精度小数转为字符串时，请保留足够的空间用于储存输出的字符串。

对 Pascal 语言的选手：

drink_sample.pas 的第 8 行和第 14 行中的“2100”为 P 的值，如果要修改请同时修改成同样的值。

请注意，按照 NOI 的有关规定，你的代码长度不能超过 100KB。