

控制流图如如上图

程序功能说明

用于计算任意多项的一元多项式加减运算，将项抽象为数对表示。

程序运行环境说明

请使在搭载windows10系统的非MAC系列电脑上JDK版本jdk-8u161-windows-x64环境中，使用eclipse-java-oxygen-2-win32-x86\_64运行该程序，run之后在Console中直接输入待计算式子(输入标准后面说明)，键入回车后在后面输出结果（输出结果后面说明）。如果以上要求不满足，运行时有可能出现未知错误。注：不支持使用文件输入。

输入格式及标准

1. 单行输入，键入回车键确认输入结束
2. 组成全部多项式的字符串中可以存在空格，所有空格直接忽略。
3. 输入仅包含数字0-9+-,(){}以及“空格”几种符号（请注意区分-的不同）。
4. 标准的输入由多组代表多项式的符号集合组成，形式如下“+{多项式1}+{多项式2}-……+{多项式n}”(前面引号中的+仅代表符号位，对于第一个多项式前面的符号可能为+-或无，对于之后的多项式前面为+-)。每组花括号内为一个多项式字符串，所有花括弧前通过‘+’或‘-’代表多项式之间的加减关系，若第一组花括号前没有则默认为‘+’，。
5. 花括号内的多项式由数对(c,n)组成，其中c为系数，n代表次数。一个多项式的标准格式为：{(c1,n1),(c2,n2),…,(cm,nm)}，最外端由花括号与其它多项式区分开，内部是若干由括号包含的数对组成，括号之间由一个逗号‘,’分隔，两个数之间也由一个‘,’分隔。一个多项式内的数对中的n都不相同。
6. 对于数对(c,n)，c为系数，为十进制整数，取值范围为-999999<=c<=999999（即c前面可以有‘-’或‘+‘）；n为该项的幂，为十进制整数，有0<=n<=999999，0可以支持+0和 -0。所有数均可能出现前导0，但c和n除去符号位之外的长度分别不超过6和6。
7. 输入时，每个多项式限制为最多50个数对，多项式的个数限制为最多20个。
8. 什么也不输入直接回车时，或者直接输入{}就回车，均算作非法输入。
9. 数对中不允许出现算式，比如(2,1-7)中，“1-7”为非法输入。

合法输入示例：

- { (3,0), (2, 2 ), (12,3) } + {(3,1), (-5,3)} - {(- 199 ,2), (29,3),(10,7)}

输出格式标准

1. 输出为一个标准的多项式表达式字符串。其语义解释方法与输入的式子的单个多项式相同。
2. 结果多项式按照多项式内的单项式次数进行升序排序。
3. 系数为0的项不会输出。
4. 如果计算结果的多项式为空（即每个项系数都是零），直接输出0
5. 任何非负数字之前不得有加号，如+3 或者+0 是被认为错误输出，-0 也是错误输出
6. 输入满足要求时，输出仅为单行，行末没有回车
7. 在输出时如不满足要求，将有说明性信息（具体如下）格式如下

ERROR

#解释

1. 如果不满足多项式式子的格式（包括多项式间缺少符号，出现非法字符，括号嵌套等等各种奇怪的非法输入。

解释为：Illegal input

1. 如果无法读入

解释为：Can't read in

1. 如果满足多项式格式但是多项式过多

解释为：Too many polynomials

1. 如果满足多项式格式，多项式<=20，但项过多

解释为：Too many polynomial Terms

1. 如果满足多项式格式，多项式<=20，项<=50，但是数字过大，无符号数字超过6位

解释为：The number is too long

1. 如果以上没有问题，但是某项指数<0

解释为：The degree should be larger than or equal to 0

1. 如果以上没有问题，但同一多项式中出现两个或多个的相同指数的项

解释为：Polynomials with items of the same degree

1. 其余奇怪的错误也进行了处理，其解释请自行探索（或直接查看代码

合法输入后的输出示例：

{(-3,0),(3,1),(197,2),(-46,3),(-10,7)}

注意事项

1. 针对正则表达式爆栈，本人已进行优化，只要不超过20\*50的规定，不会出问题。如输入过长（超过20\*50）仍有可能正则爆栈，此时即使满足多项式的格式，超过20\*50后也会直接被判为爆栈。