

1：代码部分

Cp_double_union 类

一个主模块，负责接受指令

一个函数负责进行位运算输出二进制指令

一个函数负责读入 double 并且判断是否读入正确

命令 0 退出程序

命令 1 读入 double 并输出二进制

命令 2 输出 0/0 的二进制

命令 3 输出正无穷二进制

命令 4 输出负无穷二进制

2：测试部分

1：读入

数据	数据类型	结果	正确结果
命令 1 读入 sjhf91883	开始的错误读入	报错	报错
命令 1 读入 -12.iem	之后有错误读入	和 -12 一样	和 -12 一样
命令 1 读入 12ce	之后有错误读入	和 12 一样	和 12 一样
.ce	小数点后错误读入	报错	报错
-ce	负号后错误读入	报错	报错
.128	小数点前没数字	和 0.128 一样	和 0.128 一样
-.ce	小数点和负号后错误读入	报错	报错
1929-12	负号不在第一个	和 1929 一样	和 1929 一样
0	0	正常	正常
129193	正确读入	正常	正常
123.12827	正确读入	正常	正常
-129	正确读入	正常	正常
-128.12	正确读入	正常	正常

综上：1：只需要前几位是有效 double 就好，后面的错误字符都会被屏蔽。

2：如果第一位是.，默认识别为 0.

3：如果在识别出有效 double 前就检测出非法字符，则无效

2：非数

[illegible]

在网上查到的资料：<https://blog.csdn.net/amds123/article/details/62445133>

非 数 0xFFF8000000000000, 正 无 穷 0x7FF0000000000000 , 负 无 穷 0xFFF0000000000000

可以说明正确

3：测试小数转换部分

比较方法：先把 double 转换为二进制，再把二进制转换回 double.如果两个 double 值相同，则正确

数据	类型	结果	正确与否
12013991	正整数	0100000101100110111010100011010011100000000000000000000000000000	正确
1298137317137	长整数	010000100111001011100011111100000011011111110001000100000000000000	正确
-1291992	负整数	1100000100110011101101101101100000000000000000000000000000000000	正确
-11828137717371	负长整数	11000010101001011000001111100111111001001101110011110110110000000000	正确
3.14	正小数	0100000000001001000111101011100001010001111010111000010100011111	正确
-13818.1	负小数	110000001100101011111101000011001100110011001100110011001100110011001	正确
1221.411311413143	正长	010000001001001100010101101001010010100101110110100011010111110101111	正确

	小数		
- 12.1391883813	负 长 小 数	110000000010100001000111010000111011001100010011 0101010101010111	正 确
0.3333	带 0 小 数	001111111101010101010100110010011000010111110000 0110111101101001	正 确
0	0	000 0000000000000000	错 误

0 输出的小数是 2.37152e-322，我猜测这和 double 类型的误差有关