

代码部分：

封装了 4 个输入函数：输入一个整数，输入一个有下界的整数，输入一个有上界的整数，输入一个有上下界的整数

可能的输入问题

1：长度大于 10：int 范围是-2147483648 ~ 2147483647，因此整数长度超过 10（不包括第一个负号）的字符串一定不是 int，会抛出字符串长度大小的异常

2：有非法字符：int 除了-和数字之外不可能存在其余字符，而且-至多一个，且必在首位。因此如果除首位外有非数字的字符串一定不是 int，会抛出第一个非法字符所在位置（从 1 开始计算，不包括负号）

3:0 的错误使用：一个数字（除了 0），其第一个数字位不可能是 0，会抛出 0 异常

4：超过 int 范围

5：超过限制的数字范围

6：限制不合理（下界>上界）

如果输入问题是 1-5 中一个，则会循环要求用户输入，直到合法为止

如果问题为 6，则数字不变（默认为 0）

如果没问题，则会改变数字大小，并且继续

测试部分：

1：通用（对于四种输入，不考虑超过设定范围,也就是范围设定为 int 范围）

数据	类型	正解	答案
1138381737173	长度过长的正数	报错 1,13	报错 1,13
-1138381737173	长度过长的负数	报错 1,13	报错 1,13
129197cheu13781h	长度过长的正字符	报错 1,16	报错 1,16
-129197cheu13781	长度过长的负字符	报错 1,15	报错 1,15
183717cw	非法字符	报错 2,7	报错 2,7
-183717cw	非法字符	报错 2,7	报错 2,7
-13318837.	非法字符	报错 2,9	报错 2,9
-12-18281	非法字符（负号）	报错 2,3	报错 2,3
1-12	非法字符（负号）	报错 2,2	报错 2,2
0128713	0 的错用	报错 3	报错 3
-00120123	0 的错用	报错 3	报错 3
3000000000	越界正数	报错 4	报错 4
-3000000000	越界负数	报错 4	报错 4
2147483648	越界正数（边界）	报错 4	报错 4
-2147483649	越界负数（边界）	报错 4	报错 4
2147483647	正数（边界）	2147483647	2147483647

-2147483648	正数（边界）	-2147483648	-2147483648
100	正数	100	100
-1283	负数	-1283	-1283
0	0	0	0

2：设置范围（上界 1233）

1234	越界（边界上）	报错 5	报错 5
3816313	越界	报错 5	报错 5
1233	边界	1233	1233
0	正常	0	0

3：设置范围（下界-4）

-5	越界（边界上）	报错 5	报错 5
-193183	越界	报错 5	报错 5
-4	边界	-4	-4
0	正常	0	0

4：设置范围（上界 300，下界 0）

-1	越界（边界上）	报错 5	报错 5
-193183	越界	报错 5	报错 5
301	越界（边界上）	报错 5	报错 5
131837	越界（边界上）	报错 5	报错 5
300	边界	-4	-4
0	边界	0	0

5：范围

范围 100,100，数据 100	边界	100	100
范围 100,100，数据 101	越界	报错 5	报错 5
范围 100,100，数据 99	越界	报错 5	报错 5
范围 100,99	范围不合理	报错 6	报错 6