

骗分过样例 (software)

这是一道传统题。

【题目背景】

“我的程序需要完成什么功能呀？……”

“我也不知道……”

“啊？那我怎么写呀……”

“已经有人给你写好测试了，只要你通过这些测试就可以了……”

“啊？……”

“所有的最终测试数据都在题目目录下，请做好备份，避免误删！”

“这……”

“哦，我还可以把输入格式告诉你……不过都有完整的数据了，知道输入格式可能也没太大用处吧……”

【输入格式】

从文件 *software.in* 中读入数据。

第一行输入一个字符串，表示需要运行的软件功能编号。两个编号越相似，说明对应的两个功能的算法越接近。

接下来根据功能的不同，可能有任意长度的输入，详见每个功能的文档。

【输出格式】

输出到文件 *software.out* 中。

详见每个功能的文档。

【子任务】

“‘每个功能的文档’在哪里呀？”

“我也没有，就像我没有题目描述一样……”

“好吧……那我是不是打表就可以了呀……”

“代码长度限制是 102400 字节 (100KB)，超过此限制会导致该题获得零分，直接打肯定是不行的！不过，需要的话倒是可以稍微打一些小的表……”

“唔……”

“另外，我们会给你的程序对于每个测试点分别评分，求和后得到总分。按照传统的规矩，每个测试点正确得满分，错误得 0 分。每个测试点的分值不全相同，测试点的分值、顺序与难度没有必然联系。”

测试点	功能编号	分值
1	<u>1_998244353</u>	4
2		4
3		4
4	<u>1?</u>	7
5	<u>1?+</u>	9
6	<u>1wa_998244353</u>	6
7		7
8	<u>2p</u>	4
9		6
10		8
11	<u>2u</u>	5
12		6
13		9
14	<u>2g</u>	5
15		7
16	<u>2g?</u>	9

【提示】

在你使用 C/C++ 的 int 类型时，如果发生了溢出，比较可能的情况是按照模 2^{32} 同余的前提下，在 int 范围内取一个合理的值。例如在计算 $2147483647 + 2$ 时，较有可能会得到 -2147483647 。

然而，C/C++ 标准将这种情况归类为“未定义行为”。当你的程序试图计算会溢出的 int 运算时，除了上述结果外，编译器还可能会让你的程序在此时计算出错误结果、死循环、运行错误等，这也是符合 C/C++ 标准的。

如果你的程序希望利用 int 的自然溢出的特性，请转换为 unsigned 类型运算。例如将 a + b 改写为 (int) ((unsigned) a + (unsigned) b)，以避免出现不预期的错误。