OOP 第一周作业文档

2019010175 孔瑞阳 土木 92

一、功能说明

输入一个正整数 n。
当整数 n 不是正整数的时候，提示输出的不是正整数，并要求重新输入。

输出不超过 n 的所有正偶数之和。

二、模型

1、计算方法 1

从 2 开始枚举偶数，2,4,6,...。每枚举到一个数之后加到和里面去。
当枚举到的偶数比 n 大的时候就退出循环。

2、计算方法 2

(1) 当 n 是偶数的时候，答案就是 $2+4+...+n$。

运用等差数列公式求和，答案为 $\frac{n+2}{2}*\frac{n}{2}$。

(2) 当 n 是奇数的时候，答案就是 $2+4+...+(n-1)$。

运用等差数列公式求和，答案为 $\frac{n+1}{2}*\frac{n\text{-}1}{2}$。

三、软件构件介绍

文件	功能介绍
CP_PositiveInteger.h/cpp	实现正整数的输入和操作
sumOfEvenNumberNotLargeThanN.h/cpp	计算不大于 n 的正偶数之和，以及测试
sumOfEvenNumberNotLargeThanNMain.cpp	主程序

四、单元测试

1、输入单元的测试(CP_PositiveInteger.h/cpp)

输入一个正整数:

(1) 等价类划分

① 输入的不是正整数。

② 输入的是正整数。

(2) 测试：

等价类	选取案例	输出结果（下一步操作）
等价类①	HelloWorld!	您输入的不是正整数，请输入一个正整数：
	1919.810	您输入的不是正整数，请输入一个正整数：
	-100	您输入的不是正整数，请输入一个正整数：
	0	您输入的自然数超过了 5，请输入一个自然数：
等价类②	6	第一种方法计算……（程序继续运行）
	7	第一种方法计算……（程序继续运行）
	100	第一种方法计算……（程序继续运行）

2、关于程序内容正确性的验证（ sumOfEvenNumberNotLargeThanN.h/cpp ）

等价类划分（由于非法输入已经在输入单元验证过，所以不再考虑）

n 的规模	测试方法
n<=20	手算测试
n<=92681	计算器辅助测试+对拍测试
n>=92682	特殊边界情况测试

第一步：对于比较小的数据进行手算测试。

并且可以看出，但 n 是一个偶数的时候，n 和 n+1 的答案应该是一样的。

n=6/7，则 ans = 2 + 4 + 6 = 12

n=12/13，则 ans = 2 + 4 + 6 + 8 + 10 + 12 = 42

测试结果：

输入	第一种方法	第二种方法
6	12	12
7	12	12
12	42	42
13	42	42

符合预测情况。

第二步：对于比较大的数据进行计算器测试。

输入	第一种方法	第二种方法	计算器计算
114	3306	3306	3306
514	66306	66306	66306
1919	920640	920640	920640
810	164430	164430	164430
10456	27337212	27337212	27337212
10457	27337212	27337212	27337212

两种算法都与计算器计算的结果相同。

第三步：考虑边界情况。

输入	第一种方法	第二种方法
92681	2147441940	2147441940
92682	-2147432674	-2147432674
100000	-1794917296	-1794917296

可以发现，当 n>=92682 时，由于答案已经超过了 int 的表示范围，所以出现了错误，这是由于 int 本身范围的局限性导致的。

第四步：对拍测试。

由于有两种计算方法，并且已经经过了初步测试。所以若经过多组数据验证，两种计算方法没有产生矛盾，基本可以验证程序的正确性。

经过若干分钟上十万次的对拍测试，没有发现错误。

五、关于两种计算方法的比较

第一种计算方法不需要对于 n 进行讨论,只运用了最基础的循环,所以思维难度比较小，并且不需要分类讨论。缺点是时间复杂度为 O(n)，比较高。

第二种方法运用了等差数列的计算公式，将时间复杂度优化到了 O(1)，更加快速。缺点是因为计算过程中要对于奇数和偶数分类讨论。