

《软件质量与评测技术》

单元测试实践实验报告

诚信声明

本人郑重承诺：本实验报告是本人独立工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他人已经发表或撰写过的成果。实验报告中对报告工作的任何帮助者所做的贡献均做了明确的说明并表达了谢意。

如有抄袭，本人愿意承担因此而造成任何后果。

特此声明。

签名：谢宝玛 日期：2025/12/15

任务：C 版第 3 题

实验内容与完成情况：

一， 题目

编写程序，读入一个整数 N；若 N 为非负数，则计算 N 到 $2 \times N$ 之间的整数和；若 N 为一个负数，则求 $2 \times N$ 到 N 之间的整数和。

二， 测试方法

提取出两个函数，for 循环和 while 循环，然后测试。

三， 测试用例

1. 正数测试 (for 循环) :

```
EXPECT_EQ(nsum_for(0), 0);      // 0 -> 0
EXPECT_EQ(nsum_for(1), 3);      // 1 + 2 = 3
EXPECT_EQ(nsum_for(2), 9);      // 2 + 3 + 4 = 9
EXPECT_EQ(nsum_for(3), 18);     // 3 + 4 + 5 + 6 = 18
```

2. 负数测试 (for 循环) :

```
EXPECT_EQ(nsum_for(-1), -3);    // -2 + -1 = -3
EXPECT_EQ(nsum_for(-2), -9);    // -4 + -3 + -2 = -9
EXPECT_EQ(nsum_for(-3), -18);   // -6 + -5 + -4 + -3 = -18
```

3. 正数测试 (while 循环)

```
EXPECT_EQ(nsum_while(0), 0);
EXPECT_EQ(nsum_while(1), 3);
EXPECT_EQ(nsum_while(2), 9);
EXPECT_EQ(nsum_while(3), 18);
```

4. 负数测试 (while 循环)

```
EXPECT_EQ(nsum_while(-1), -3);
```

```
EXPECT_EQ(nsum_while(-2), -9);
```

```
EXPECT_EQ(nsum_while(-3), -18);
```

5. 大数测试（防止边界循环错误）

```
EXPECT_EQ(nsum_for(1000), nsum_while(1000));
```

```
EXPECT_EQ(nsum_for(-1000), nsum_while(-1000));
```

6. 一致性测试（核心测试）

```
for (int n = -100; n <= 100; ++n) {  
  
    EXPECT_EQ(nsum_for(n), nsum_while(n))  
  
    << "Mismatch at n = " << n;  
  
}
```

7. 数学正确性测试（公式验证）

```
int n = 10;  
  
long expected = (10 + 20) * 11 / 2; // 等差数列求和  
  
EXPECT_EQ(nsum_for(n), expected);  
  
EXPECT_EQ(nsum_while(n), expected);
```

四，结果

七个方向的所有测试用例均通过。

测试结果截图：

```
[=====] Running 7 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 7 tests from NSumTest
[ RUN    ] NSumTest.ForLoopPositive
[      OK ] NSumTest.ForLoopPositive (0 ms)
[ RUN    ] NSumTest.ForLoopNegative
[      OK ] NSumTest.ForLoopNegative (0 ms)
[ RUN    ] NSumTest.WhileLoopPositive
[      OK ] NSumTest.WhileLoopPositive (0 ms)
[ RUN    ] NSumTest.WhileLoopNegative
[      OK ] NSumTest.WhileLoopNegative (0 ms)
[ RUN    ] NSumTest.LargeNumberTest
[      OK ] NSumTest.LargeNumberTest (0 ms)
[ RUN    ] NSumTest.ConsistencyTest
[      OK ] NSumTest.ConsistencyTest (0 ms)
[ RUN    ] NSumTest.FormulaValidation
[      OK ] NSumTest.FormulaValidation (0 ms)
[-----] 7 tests from NSumTest (0 ms total)

[-----] Global test environment tear-down
[=====] 7 tests from 1 test suite ran. (0 ms total)
[ PASSED ] 7 tests.
```

出现的问题：

一， C 与 C++ 程序的兼容性问题

测试的程序是 C 语言编写， googletest 框架是 C++ 编写，所以引入头文件的时候会出现不兼容的问题。

二， 程序一体化的问题

最开始的程序是一个只包含 main 函数的程序，不方便测试。

解决方案（列出遇到的问题和解决办法，列出没有解决的问题）：

一， C 与 C++ 程序的兼容性问题

通过 extern “C” 可以导入 C 语言头文件。

二， 程序一体化的问题

抽象出两个函数：

// 计算从 n 到 2*n 的和（使用 for 循环）

long nsum_for(int n);

// 计算从 n 到 2*n 的和（使用 while 循环）

long nsum_while(int n);

参考资料：

googletest 官方文档: <https://google.github.io/googletest/>

致谢：

单纯老师， Chat-GPT。

