

测试报告

第 25 组

2022 年 3 月 22 日

1 概述

本次测试中，执行了所有设计的测试用例。**测试用例所有执行结果均以截图的形式保存在附件中。**

2 对各功能测试的结果

2.1 十六进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5 以外的所有测试用例。

在执行用例 5 时，输出的结果与预期不符，但是与用例 6 完全一致。后面又设计了两个略低于最大值的输入（0x3fffffffffff3, 0x3ffffffffffe7），发现输出依然未变。经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

2.2 三十二进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5 以外的所有测试用例。

在执行用例 5 时，输出的结果与预期不符，但是与用例 6 完全一致。经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

2.3 六十四进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5 以外的所有测试用例。

在执行用例 5 时，输出的结果与预期不符，但是与用例 6 完全一致。经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

2.4 二进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5，例 6，例 7 以外的所有测试用例。

在执行用例 5 时，输出的结果与预期不符。经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

2.5 四进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5，例 6，例 7 以外的所有测试用例。

在执行用例 5，例 6，例 7 时，输出的结果与预期不符，经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

2.6 八进制转换功能的测试结果

在对该功能进行测试的过程中，通过了除用例 5，例 6，例 7 以外的所有测试用例。

在执行用例 5，例 6，例 7 时，输出的结果与预期不符，经过对源代码的分析，发现该问题是由浮点运算损失精度导致。

3 覆盖率分析

3.1 需求覆盖率

由于仅仅是对五个功能进行用例设计和测试，所以也只覆盖了对应的五个功能的需求。

3.2 代码覆盖率

基于同样的原因，本次测试仅覆盖了实现五个功能的 javascript 代码，未覆盖网页其他代码。

4 结论和建议措施

五个功能均能正确实现小整数进制转换。但是当输入变大时，会由于浮点数精度损失导致结果错误。建议改用整数类型进行计算，即可避免该问题。

5 各小组成员贡献比例

谭博仁、周峰和张贻天贡献均为 33%，KENT HUGO GEMIARTHA 贡献为 0。