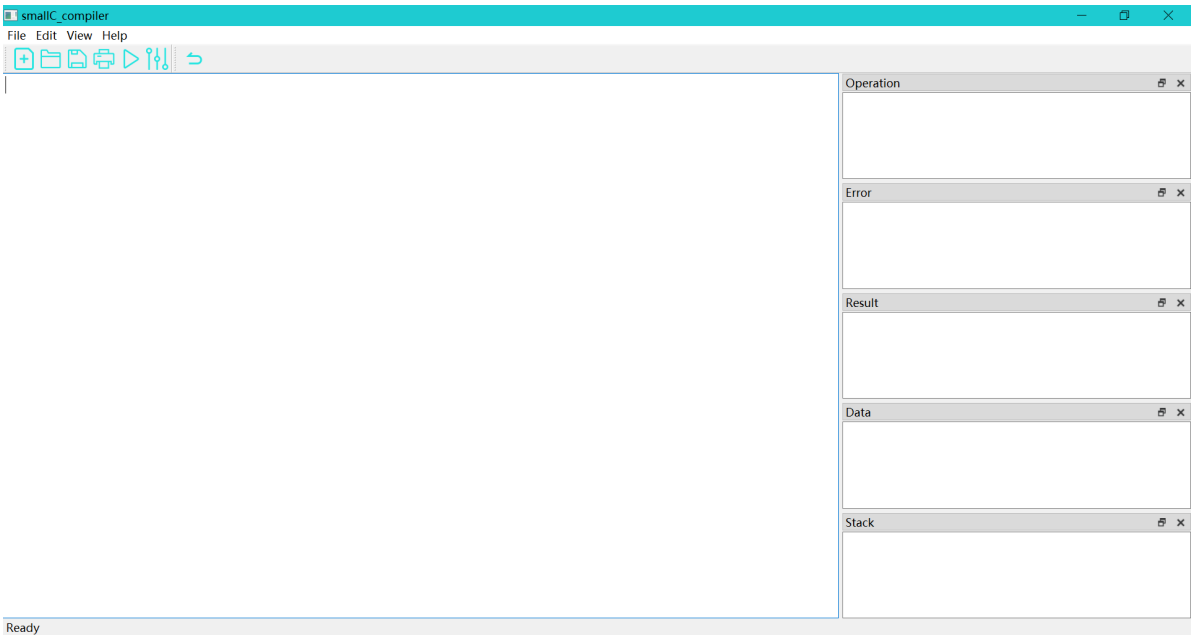
smallC-programming-language-compiler 测试说明书

1 概述

1.1 测试描述

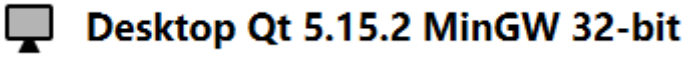
1.1.1 界面结构



1.1.2 功能介绍

1.2 测试环境

`Qt 5.15.2`



2 测试用例描述

2.1 测试一

1. 测试目标

输出100以内所有的素数

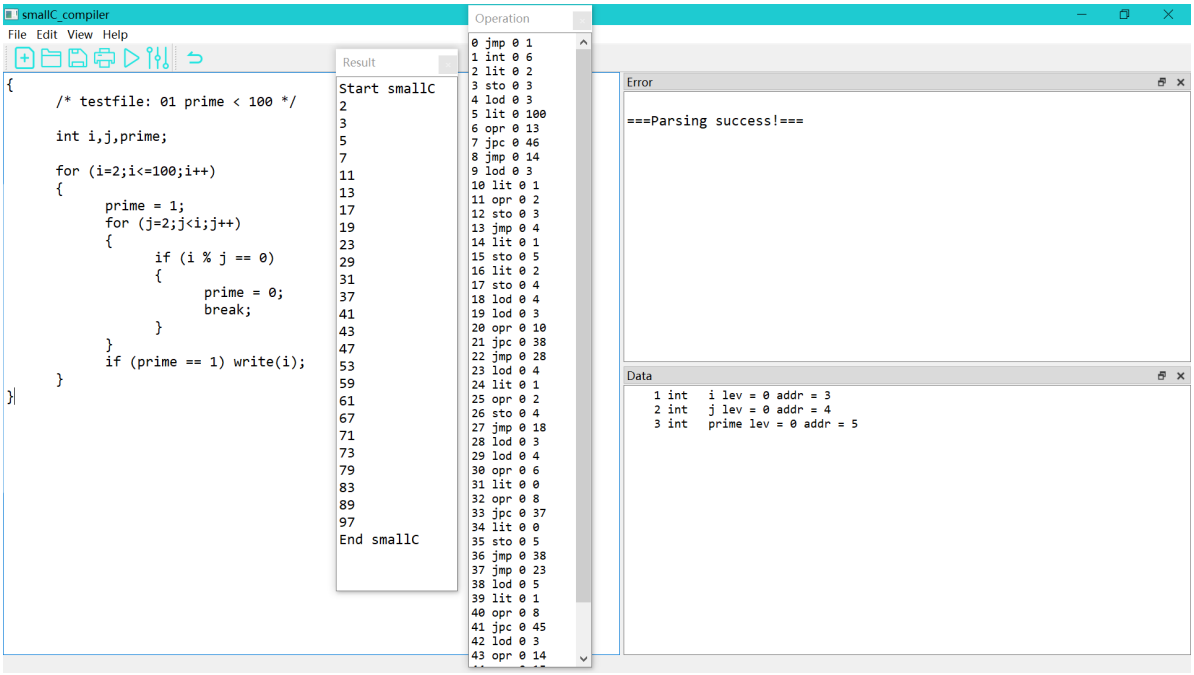
2. 测试源程序

```
{
    /* testfile: 01 prime < 100 */

    int i,j,prime;

    for (i=2;i<=100;i++)
    {
        prime = 1;
        for (j=2;j<i;j++)
        {
            if (i % j == 0)
            {
                prime = 0;
                break;
            }
        }
        if (prime == 1) write(i);
    }
}
```

3. 测试步骤

点击`open file`键，打开测试文件`test_01_prime.txt`，点击`run`键

4. 测试结果



2.2 测试二

1. 测试目标

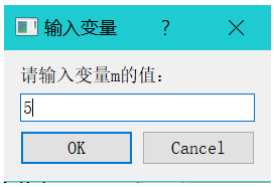
输出两个数的最小公倍数

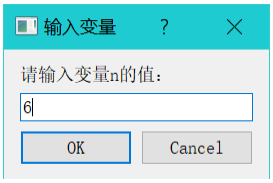
2. 测试源程序

```
{
    /* testfile: 02 lcm */

    int m,n,x,y,temp,lcm,gcd;

    read(m,n);
    x = m;
    y = n;
    while(y)
    {
        temp = x % y;
        x = y;
        y = temp;
    }
    gcd = x + y;
    lcm = m * n / gcd;
    write(lcm);
}
```

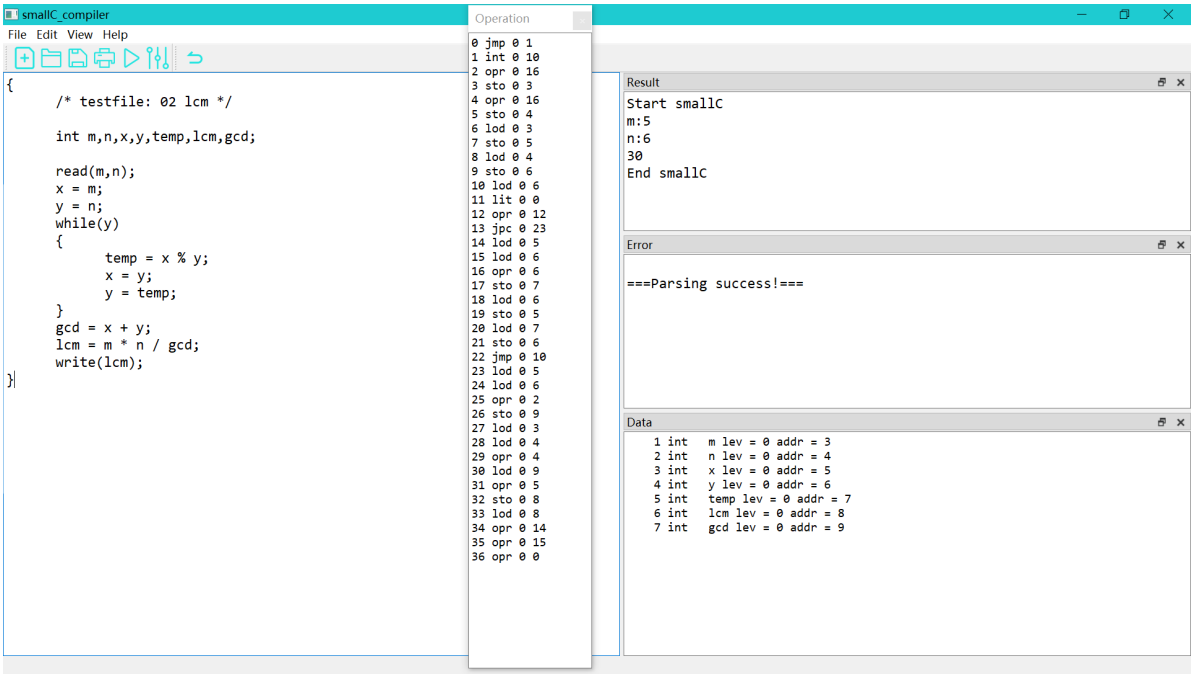
3. 测试步骤

点击`open file`键，打开测试文件`test_02_lcm.txt`，分别输入m和n的值，点击`run`键





4. 测试结果



2.3 测试三

1. 测试目标

if-else语句与单步调试

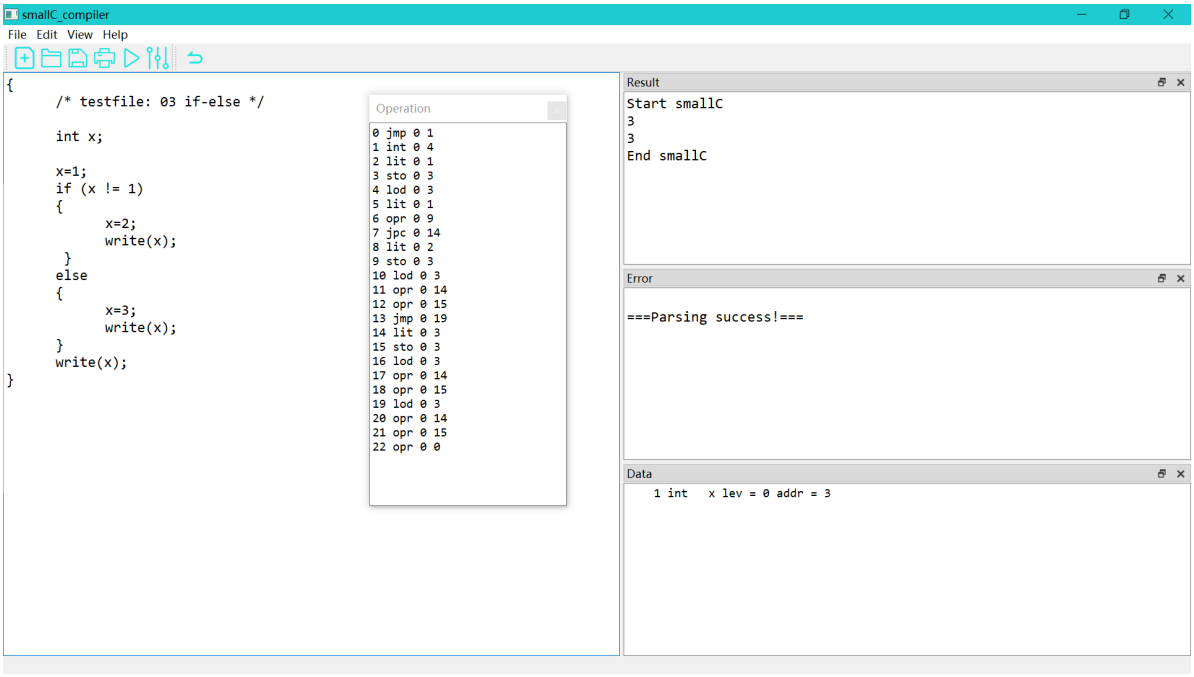
2. 测试源程序

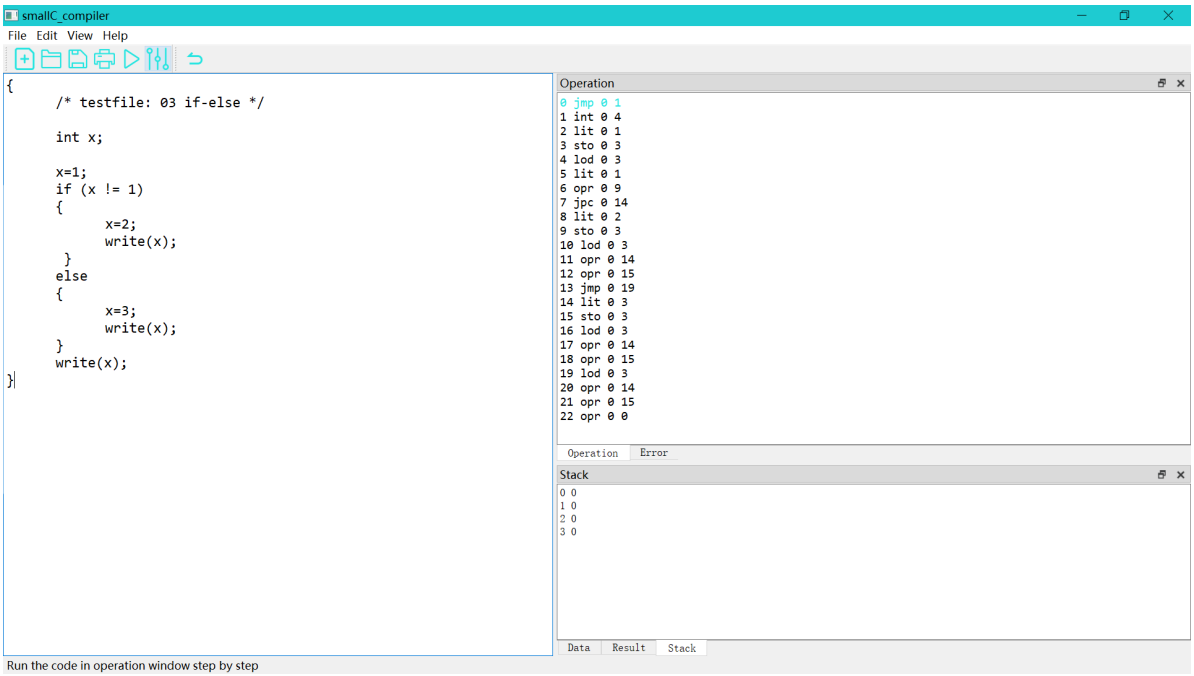
```
{
    /* testfile: 03 if-else */

    int x;

    x=1;
    if (x != 1)
    {
        x=2;
        write(x);
     }
    else
    {
        x=3;
        write(x);
    }
    write(x);
}
```

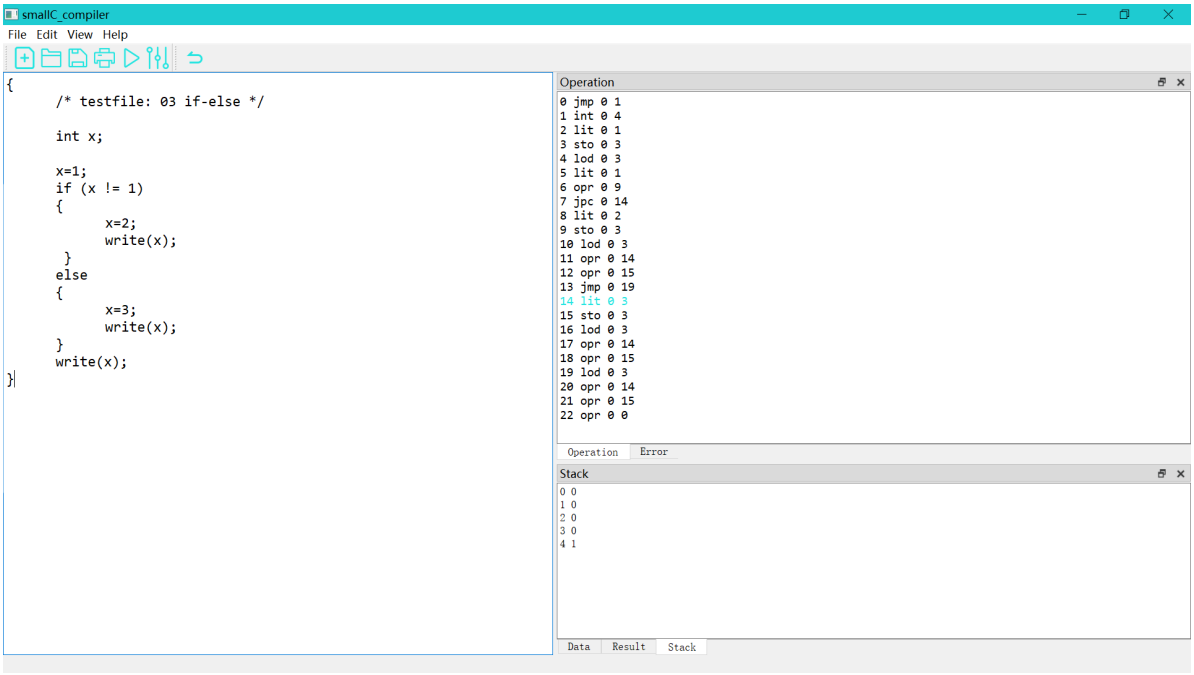
3. 测试步骤

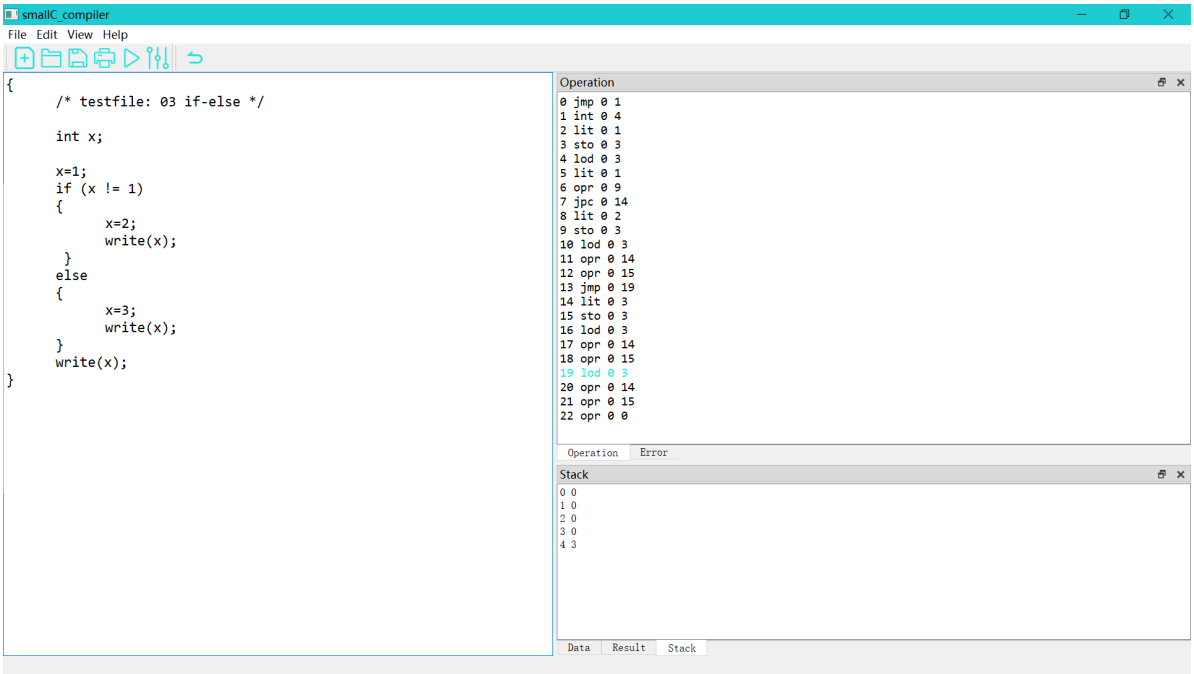
点击`open file`键，打开测试文件`test_03_if.txt`，点击`run`键，不断点击`step-run`进行单步调试

4. 测试结果









2.4 测试四

1. 测试目标

xor运算

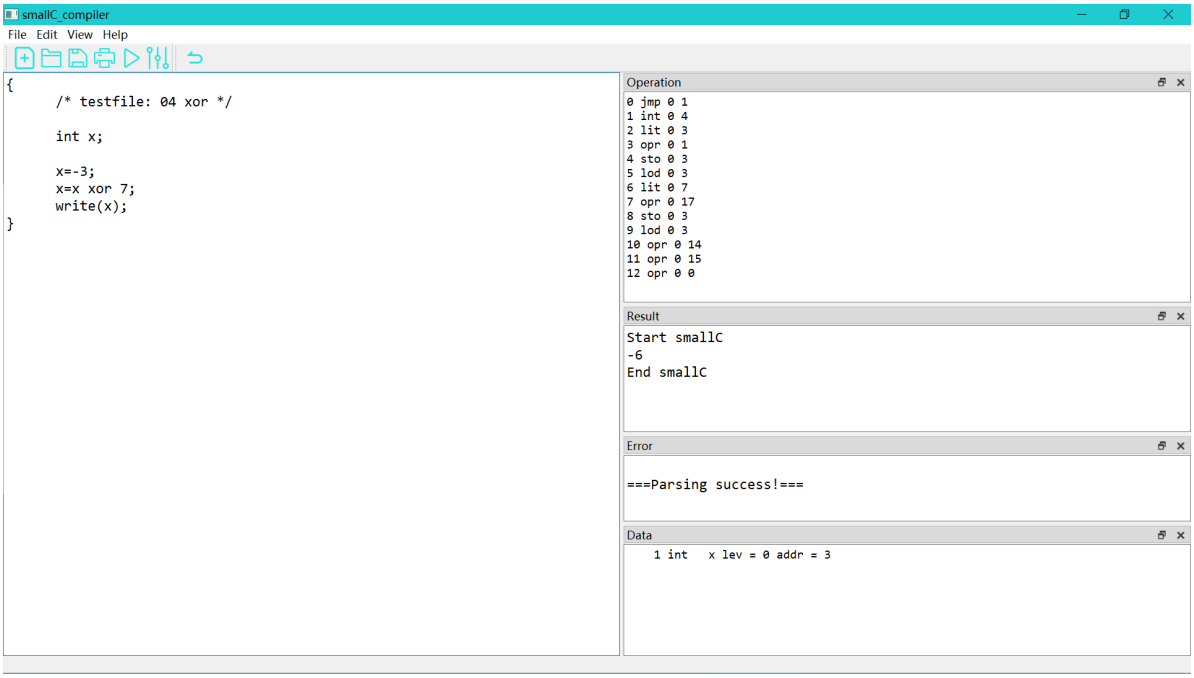
2. 测试源程序

```
{
    /* testfile: 04 xor */

    int x;

    x=-3;
    x=x xor 7;
    write(x);
}
```

3. 测试步骤

点击`open file`键，打开测试文件`test_04_xor.txt`，点击`run`键

4. 测试结果



2.5 测试五

1. 测试目标

do-while语句

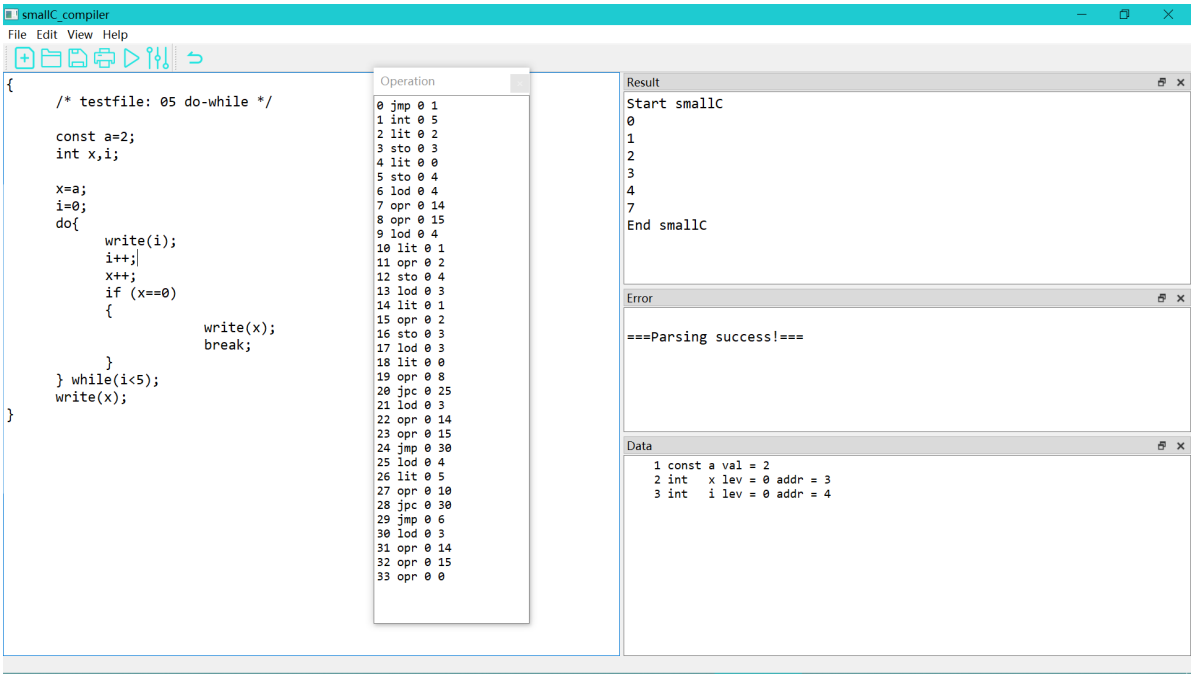
2. 测试源程序

```
{
    /* testfile: 05 do-while */

    const a=2;
    int x,i;

    x=a;
    i=0;
    do{
        write(i);
            i++;
            x++;
            if (x==0)
            {
                write(x);
                    break;
            }
    } while(i<5);
    write(x);
}
```

3. 测试步骤

点击`open file`键，打开测试文件`test_05_dowhile.txt`，点击`run`键

4. 测试结果



2.6 测试六

1. 测试目标

repeat-until语句

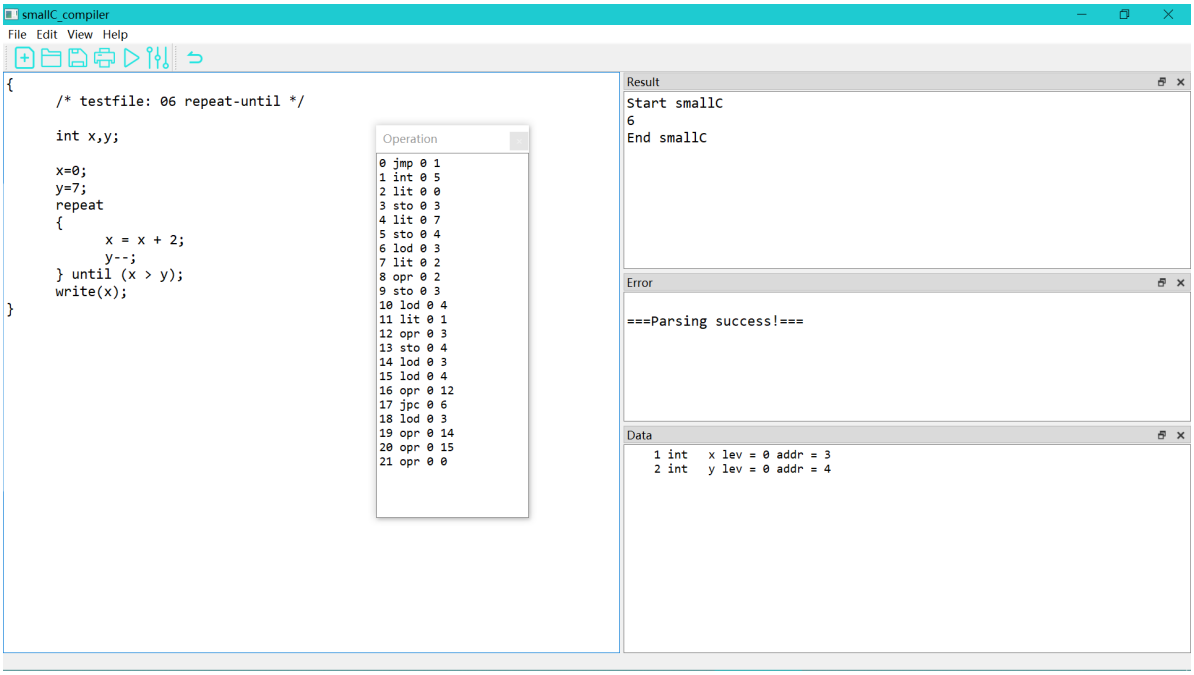
2. 测试源程序

```
{
    /* testfile: 06 repeat-until */

    int x,y;

    x=0;
    y=7;
    repeat
    {
        x = x + 2;
        y--;
    } until (x > y);
    write(x);
}
```

3. 测试步骤

点击`open file`键，打开测试文件`test_06_repeatuntil.txt`，点击`run`键

4. 测试结果



2.7 测试七

1. 测试目标

出错处理

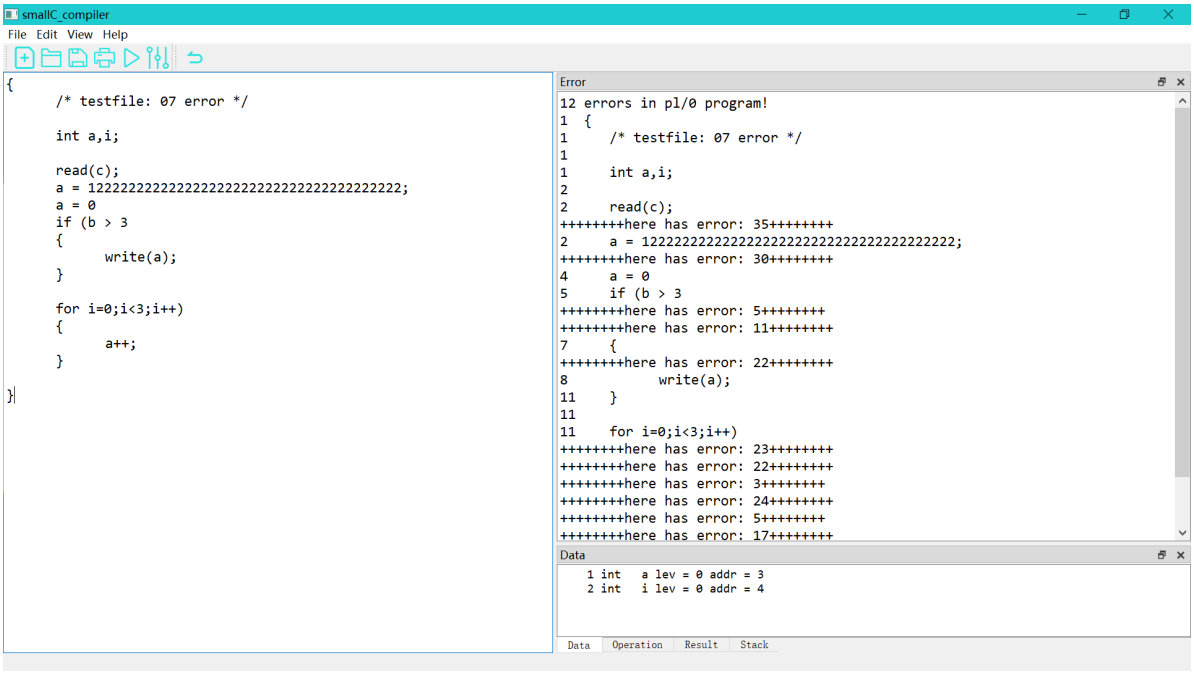
2. 测试源程序

```
{
    /* testfile: 07 error */

    int a,i;

    read(c);
    a = 12222222222222222222222222222222222222;
    a = 0
    if (b > 3
    {
        write(a);
    }

    for i=0;i<3;i++)
    {
        a++;
    }

}
```

3. 测试步骤

点击`open file`键，打开测试文件`test_07_error.txt`，点击`run`键

4. 测试结果


smallC_compiler
File Edit View Help
{
/* testfile: 07 error */
int a,i;
read(c);
a = 12222222222222222222222222222222222222;
a = 0
if (b > 3
{
write(a);
}
for i=0;i<3;i++)
{
a++;
}
}
Error
12 errors in pl/0 program!
1 {
1 /* testfile: 07 error */
1
1 int a,i;
2
2 read(c);
++++++++here has error: 35++++++++
2 a = 12222222222222222222222222222222222222;
++++++++here has error: 30++++++++
4 a = 0
5 if (b > 3
++++++++here has error: 5++++++++
++++++++here has error: 11++++++++
7 {
++++++++here has error: 22++++++++
8 write(a);
11 }
11
11 for i=0;i<3;i++)
++++++++here has error: 23++++++++
++++++++here has error: 22++++++++
++++++++here has error: 3++++++++
++++++++here has error: 24++++++++
++++++++here has error: 5++++++++
++++++++here has error: 17++++++++
Data
1 int a lev = 0 addr = 3
2 int i lev = 0 addr = 4
Data
Operation
Result
Stack