Testing Bradley's Greatest Common Divisor Program on EXPER

Robert L. Hess and Frederick G. Sayward
Research Report #165

May 1979

This research was supported in part by the Army Institute for Research in Management Information and Computer Science and the Army Research Office under Grant DAAG 29-78-G-0121 to the Georgia Institute of Technology School of Information and Computer Science, subcontracted to Yale University.
Testing Bradley’s Greatest Common Divisor Program on EXPER

Robert L. Hess
and
Frederick G. Sayward

Department of Computer Science
Yale University
New Haven, Connecticut 06520

May 1979

ABSTRACT

A program testing experiment run on the EXPER program testing system is described. The experiment involved taking a published generalization of Euclid’s greatest common divisor algorithm which was known to be incorrect and seeing if trying to pass the mutation test would lead to discovering the errors. All errors were uncovered and in addition a slight improvement to the algorithm was found.

INTRODUCTION

The EXPER system [1] is an experimental program testing system designed by Timothy Budd, Richard DeMillo, Richard Lipton, and Frederick Sayward. In this report we will present experiences gained while analyzing a fast generalization of Euclid’s greatest common divisor algorithm on EXPER. The algorithm was introduced and its complexity analyzed in [2,3]. A brief description follows:

Input: a natural number n>0 and integers A[1], ..., A[n].

Bradley's algorithm is of particular interest to us because a few years after its publication, during an attempt to formally prove it correct [4], several subtle errors were found to be present. Hence, if we start with the original program presented in [2,3] and run it through EXPER, we should expect to uncover at least the errors detected in [4].

**METHOD**

We began running EXPER on Bradley's Algorithm as it appeared in [3], called Algorithm 386. See appendix A for the initial program listing. Note that there is more than one correct answer to any input, since it is not guaranteed that the program will output a minimal set of multipliers. However, we decided to ignore this and let mutant correctness be determined by exact equality with the program. No mutant was allowed to run more than ten times longer than the original program.

The following is a history of our constructing test cases on EXPER which caused Bradley's Algorithm to fail and the corrective action we took. The line numbers refer to the corrected version of Bradley's Algorithm which is listed in appendix B.

The random test case 0 3 9 caused I to be undefined in line 52. The correction is the addition of line 48: I=N.

The cleanup loop (lines 62-67) is used when IGCD is found to be one. To test this portion of code we input -1 3. The result was K undefined in line 55; this was corrected by adding the condition statement at lines 50-51: IF(MP2.GT.1) GOTO 51.
On re-running with -1 3 the algorithm gave an incorrect answer. This was corrected by adding line 63-64: IF(IPl.GT.N) GOTO 40.

0 0 0 -3 was a special case which tested lines 1-11. The program gave an incorrect answer. This error was fixed by changing line 9 to IGCD=IABS(A(M)) and changing line 10 to Z(M)=A(M)/IGCD.

The test case 0 42 -6 15 exercised the gcd and multiplier loops (lines 20-61) with M equal to 2. This value of M caused K to be undefined in line 55. This was corrected by changing line 53 to K=I-J+MPl.

All of these changes were noted in the Certification of Algorithm 386 [4]. We found no additional errors.

ANALYSIS

Appendix B lists the final EXPER report on the corrected version of Bradley’s Algorithm. There are two reasons for the high number of equivalent mutants. First, quite a few mutations are trivial: replacing Y1 by the absolute value of Y1 is equivalent because Y1 is always positive. Second, line 45-46: IF(Cl.EQ.1) GOTO 60 is not an essential statement for the program’s functional correctness. It’s only purpose is to reduce the number of calculations if the greatest common divisor is found to be 1.

Through this experiment, we found one improvement that can be made to the algorithm. Line 49: IGCD=A(M) can be replaced by IGCD=Cl, which is slightly faster because it does not reference an array.

REFERENCES

[1] Tim Budd, Richard DeMillo, Richard Lipton, and Frederick Sayward, "Mutation Analysis", Yale University Department of Computer Science
Research Report 155, April 1979, pp. 28.


APPENDIX A

The following is a listing of Bradley’s original algorithm for calculating the greatest common divisor of n integers and multipliers as found in [3].

SUBROUTINE GCDN(N,A,Z,IGCD)
C
C    N    NUMBER OF INTEGERS
C    A() INPUT ARRAY OF N INTEGERS.  INPUT IS DESTROYED.
C    Z() OUTPUT ARRAY OF N MULTIPLIERS
C    IGCD OUTPUT GREATEST COMMON DIVISOR
C
INPUT A
RDONLY N
OUTPUT Z,IGCD
DIMENSION A(N),Z(N)
INTEGER A,Z,C1,C2,Y1,Y2,Q
C
C    FIND THE FIRST NON-ZERO INTEGER
    DO 1 M=1,N,1
       IF(A(M).NE.0) GOTO 3
  1 Z(M)=0
    2 3
C    ALL ZERO INPUTS RESULTS IN ZERO GCD AND Z
    IGCD=0
    RETURN
  4 5
C    IF LAST NUMBER IS THE ONLY NON-ZERO NUMBER, EXIT IMMEDIATELY
    IF(M.NE.N) GOTO 4
    7 8
    IGCD=A(M)
    Z(M)=1
    RETURN
  9 10
    M1=M+1
    MP2=M+2
 11 12
C    CHECK THE SIGN OF A(M)
    ISIGN=0
    IF(A(N).GE.0) GOTO 5
 14 15 16
    ISIGN=1
    A(M)=-A(M)
 17 18
C    CALCULATE GCD VIA N-1 APPLICATIONS OF THE GCD ALGORITHM FOR TWO INTEGERS.
C    SAVE THE MULTIPLIERS.
 19 DO 30 I=M1,N,1
    IF(A(I).NE.0) GOTO 7
    21 22
    A(I)=1
    Z(I)=0
    GOTO 25
 23 24 25
    Y1=1
    Y2=0
    C2=1ABS(A(I))
 26 27 28
    Q=C2/C1
    C2=C2-Q*C1
 29 30
C    TESTING BEFORE COMPUTING Y2 AND BEFORE COMPUTING Y1 BELOW SAVES N-1
C    ADDITIONS AND N-1 MULTIPLICATIONS
IF(C2.EQ.0) GOTO 20
Y2=Y2-Q*Y1
Q=C1/C2
C1=C1-Q*C2
IF(C1.EQ.0) GOTO 15
Y1=Y1-Q*Y2
GOTO 10
15 C1=C2
Y1=Y2
20 Z(I)=(C1-Y1*A(M))/A(I)
A(I)=Y1
A(M)=C1
C TERMINATE GCD CALCULATIONS IF GCD EQUALS 1
25 IF(C1.EQ.1) GOTO 60
30 CONTINUE
40 I=I+1
C CALCULATE MULTIPLIERS
DO 50 J=MP2,I,1
K=I-J+2
KK=K+1
Z(K)=Z(K)*A(KK)
50 A(K)=A(K)*A(KK)
51 Z(M)=Z(MP1)
IF(ISIGN.EQ.0) GOTO 100
58 Z(M)=-Z(M)
100 RETURN
C GCD FOUND, SET REMAINDER OF THE MULTIPLIERS EQUAL TO ZERO.
60 MP1=I+1
DO 65 J=MP1,N,1
65 Z(J)=0
GOTO 40
END
APPENDIX B

In this appendix we list the corrected version of Bradley's algorithm and the final EXPER report which contains the test cases and an accounting of the algorithm's mutants. Asterisks denote corrections.

SUBROUTINE GCDN(N,A,Z,IGCD)
C
C N NUMBER OF INTEGERS
C A() INPUT ARRAY OF N INTEGERS. INPUT IS DESTROYED.
C Z() OUTPUT ARRAY OF N MULTIPLIERS
C IGCD OUTPUT GREATEST COMMON DIVISOR
C
INPUT A
RONLY N
OUTPUT Z,IGCD
DIMENSION A(N),Z(N)
INTEGER A,Z,C1,C2,Y1,Y2,Q
C
C FIND THE FIRST NON-ZERO INTEGER
DO 1 M=1,N,1
   IF(A(M).NE.0) GOTO 3
1   Z(M)=0
   RETURN
C ALL ZERO INPUTS RESULTS IN ZERO GCD AND Z
IGCD=0
RETURN
C IF LAST NUMBER IS THE ONLY NON-ZERO NUMBER, EXIT IMMEDIATELY
3   IF(M.NE.N) GOTO 4
   IGCD=IABS(A(M))
   Z(M)=A(M)/IGCD
   RETURN
4   MP1=M+1
   MP2=M+2
C CHECK THE SIGN OF A(M)
   ISIGN=0
   IF(A(M).GE.0) GOTO 5
   ISIGN=1
   A(M)=-A(M)
5   C1=A(M)
   DO 30 I=MP1,N,1
      IF(A(I).NE.0) GOTO 7
   30   A(I)=1
   Z(I)=0
   GOTO 25
7   Y1=1
   Y2=0
   C2=IABS(A(I))
   Q=C2/C1
   C2=C2-Q*C1
   GOTO 3
25   IGCD=IGCD+1
   RETURN
C TESTING BEFORE COMPUTING Y2 AND BEFORE COMPUTING Y1 BELOW SAVES N-1
C ADDITIONS AND N-1 MULTIPLICATIONS
   IF(C2.EQ.0) GOTO 20
   Y2=Y2-Q*Y1
   Q=C1/C2
   C1=C1-Q*C2
   IF(C1.EQ.0) GOTO 15
   Y1=Y1-Q*Y2
   GOTO 10
20   C1=C2
   Y1=Y2
   Z(I)=(C1-Y1*A(M))/A(I)
   A(I)=Y1
   A(N)=C1
C TERMINATE GCD CALCULATIONS IF GCD EQUALS 1
25   IF(C1.EQ.1) GOTO 60
30   CONTINUE
40   IGCD=A(M)
C CALCULATE MULTIPLIERS
   IF(MP2.GT.1) GOTO 51
   DO 50 J=MP2,I,1
   K=I-J+MP1
   KK=K+1
   Z(K)=Z(K)*A(KK)
50   A(K)=A(K)*A(KK)
   Z(M)=A(MP1)
   IF(ISIGN.EQ.0) GOTO 100
   Z(M)=-Z(M)
100   RETURN
C GCD FOUND, SET REMAINDER OF THE MULTIPLIERS EQUAL TO ZERO.
60   IP1=I+1
   IF(IP1.GT.N) GOTO 40
   DO 65 J=IP1,N,1
65   Z(J)=0
   GOTO 40
END
The following 34 test cases were developed to kill mutants.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 IN:</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2 IN:</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3 IN:</td>
<td>4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4 IN:</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5 IN:</td>
<td>2</td>
<td>-1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6 IN:</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>7 IN:</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8 IN:</td>
<td>2</td>
<td>36</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9 IN:</td>
<td>4</td>
<td>36</td>
<td>54</td>
<td>12</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>10 IN:</td>
<td>2</td>
<td>36</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11 IN:</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12 IN:</td>
<td>4</td>
<td>36</td>
<td>90</td>
<td>5000</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>2</td>
<td>2222</td>
<td>-1111</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13 IN:</td>
<td>3</td>
<td>0</td>
<td>-3</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14 IN:</td>
<td>3</td>
<td>-5</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>5</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15 IN:</td>
<td>3</td>
<td>-8</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16 IN:</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17 IN:</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>14</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>OUT:</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>------</td>
<td>---</td>
<td>-----</td>
<td>------</td>
<td>---</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>6</td>
<td>318</td>
<td>-19</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>3</td>
<td>300</td>
<td>207</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>3</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>6</td>
<td>2</td>
<td>102</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>42</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>1</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>1</td>
<td>-2</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>2</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>2</td>
<td>6</td>
<td>318</td>
<td>-19</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>3</td>
<td>300</td>
<td>207</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>3</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>6</td>
<td>2</td>
<td>102</td>
<td>30</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>42</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>1</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>1</td>
<td>-2</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>14</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>2</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RESULTS:

NUMBER OF TEST CASES = 34

NUMBER OF MUTANTS = 5121
NUMBER OF DEAD MUTANTS = 4956 (96.8%)
NUMBER OF LIVE MUTANTS = 2 (0.0%)
NUMBER OF EQUIV MUTANTS = 163 (3.2%)

NUMBER OF MUTABLE STATEMENTS = 67
GIVING A MUTANTS/STATEMENT RATIO OF 76.43

<table>
<thead>
<tr>
<th>MUTANT TYPE</th>
<th>TOTAL</th>
<th>DEAD</th>
<th>LIVE</th>
<th>EQUIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT REPLACEMENT</td>
<td>42</td>
<td>39</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SCALAR VARIABLE REPLACEMENT</td>
<td>1575</td>
<td>1562</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>SCALAR FOR CONSTANT REP.</td>
<td>336</td>
<td>314</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>CONSTANT FOR SCALAR REP.</td>
<td>222</td>
<td>220</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>SOURCE CONSTANT REPLACEMENT</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ARRAY REF. FOR CONSTANT REP.</td>
<td>180</td>
<td>176</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>ARRAY REF. FOR SCALAR REP.</td>
<td>900</td>
<td>899</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>COMPARIALLE ARRAY NAME REP.</td>
<td>30</td>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CONSTANT FOR ARRAY REF REP.</td>
<td>51</td>
<td>51</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SCALAR FOR ARRAY REF REP.</td>
<td>480</td>
<td>479</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ARRAY REF. FOR ARRAY REF.</td>
<td>240</td>
<td>239</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>UNARY OPERATOR INSERION</td>
<td>463</td>
<td>359</td>
<td>1</td>
<td>103</td>
</tr>
<tr>
<td>ARITHMETIC OPERATOR REPLA</td>
<td>154</td>
<td>154</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RELATIONAL OPERATOR REPLA</td>
<td>50</td>
<td>43</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>UNARY OPERATOR REMOVAL</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STATEMENT ANALYSIS</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STATEMENT DELETION</td>
<td>66</td>
<td>65</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RETURN STATEMENT REPLACEMEN</td>
<td>63</td>
<td>63</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GOTO STATEMENT REPLACEMEN</td>
<td>195</td>
<td>191</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>DO STATEMENT END REPLACEMEN</td>
<td>30</td>
<td>28</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
The following mutants remain live since we did not find a test case to kill them and it is not obvious that they are equivalent.

**STATEMENT 45 CHANGED FROM**

25 IF(C1.EQ.1) GOTO 60

**TO**

25 IF(C1.EQ.Z(I)) GOTO 60

**STATEMENT 42 CHANGED FROM**

20 Z(I)=(C1-Y1*A(M))/A(I)

**TO**

20 Z(I)=(C1-Y1*++A(M))/A(I)