CHAPTER 8

TRACE TABLES

Trace Tables

8.1 Introduction

One of the steps in the program development process is the testing of the final program design. To do this, we perform a "dry run" of the pseudocde. A **dry run** is the process by which we select appropriate test data and perform an execution of the algorithm by hand using the test data until the program terminates. This either produce the desired results or the program fails to achieve the desired purpose. In the latter case, the program logic is examined to determine the problem. When performing a dry run of a program, the state of the variables used by the program is updated after each program instruction is carried out. When the program terminates, the correct data should be printed and/or the variables used by the program contain the correct final results. The state of the variable is stored in a **trace table.**

8.2 Trace Tables

A trace table is a rectangular array where the column heading records the names of all the variables used in the algorithm and the rows record the state of the variables after every instruction executes.

PRACTICE EXAMPLE 29

Use trace table to test the accuracy of the logic of the following pseudocode:

```
Sum = 0
read Number
While number <> 0 do
    Sum = Sum + Number
    read Number
endwhile
print "Sum of number is", Sum
```

use the following test data as input:

12, 23, 34, 0

Sum	Number
0	

Figure 8.1

Sum	Number
0	12

Figure 8.2

Sum	Number
0	12
12	

Figure 8.3

Sum	Number
0	12
12	23

Figure 8.4

SOLUTION

The first instruction in the pseudocode is:

Sum = 0

When this statement is carried out, the trace table is as shown in Figure 8.1. The second instruction in the pseudocode is:

read Number

The first number in the sequence of input is 12. When this instruction is executed, the trace table is as shown in Figure 8.2. The condition statement is tested:

Number <> 0

We enter the loop because the content of Number is 12. The first instruction inside the loop is:

Sum = Sum + Number

The variable Sum now contains the value 12 (0 + 12). The trace table is as shown in Figure 8.3 The next statement to be executed is:

read Number

After this statement is executed the state of the trace table is as shown in Figure 8.4 because the next value in the input sequence is 23.

The condition Number <> 0 is tested. The loop block is executed again because our condition is still true. The next statement to be executed is:

Sum = Sum + Number

The variable Sum now contains the value 35 (12 + 23). The state of the trace table is as shown in Figure 8.5 The next statement to be executed is:

read Number

Sum	Number
0	12
12	23
35	

Figure 8.5

Sum	Number
0	12
12	23
35	34

Figure 8.6

Sum	Number
0	12
12	23
35	34
69	

Figure 8.7

Sum	Number
0	12
12	23
35	34
69	0

Figure 8.8

After this statement is executed the state of the trace table is as shown in Figure 8.6 because the next value in the input sequence is is 34. The loop block is executed another time because the content of Number is still not 0. The next statement to be executed is:

Sum = Sum + Number

The variable Sum now contains the value 69 (35 + 34). The trace table is as shown in Figure 8.7 The next statement to be executed is:

read Number

The next Number read is 0. The loop terminates because the content of Number is 0 and the conditional statement:

Number <> 0

becomes false. The next instruction to be executed is:

print "Sum of numbers is ", SUM

This instruction produces the following output on the computer screen:

Sum of numbers is 69

The Program comes to an end. The final state of the variable in our pseudocode is shown in the last row of the trace table in Figure 8.8. The final content of Sum is 69, which indicates that the logic of our program is correct.

NCount	ZCount	PosSum	Number

Figure 8.9

PRACTICE EXAMPLE 30

Test the logics of the pseudocode below for correctness using a trace table. Use the following as your test data:

-12, 23, 30, 0, -120, 50, 0, 0, 15, 999

```
NCount = 0
ZCount = 0          statements to
PosSum = 0          initialize count

read Number
while Number <> 999 do
    if Number < 0 then
        NCount = NCount + 1         If statement to test if
    endif                           Number is less than
                                    zero
    if Number <> 0 then
        ZCount = ZCount + 1         If test statement to
    endif                           test if number is zero (0).

    if Number > 0 then
        PosSum = PosSum + Number    If statement to test
    endif                           if number is positive
endwhile

print  "Number of negative numbers is ", NCount
print  "Number of zeros is ", ZCount          Statement to print
print  "Sum of positive numbers is", PosSum   content of variables
```

SOLUTION

Figure 8.9 shows the layout of the trace table for the pseudocode. The four variables used in the pseudocode (NCount, ZCount, PosSum and Number) forms the table headings.

NCount	ZCount	PosSum	Number
0	0	0	-12

Figure 8.10

NCount	ZCount	PosSum	Number
0	0	0	-12
1			

Figure 8.11

The first four statements in the pseudoode are:

NCount = 0
ZCount = 0
PosSum = 0

read Number

After the four statements above are executed, the state of the trace table is as shown in Figure 8.10. The loop condition (Number <> 999) is tested. If the condition is true, we enter the loop. The first test on the condition is performed, the content of Number is 12. The test is true therefore the statement:

NCount = NCount + 1

is executed. The value of NCount becomes 1 and the other tests are ignored because they are false. The state of the trace table is as shown in Figure 8.11. The next statement to be performed is

read Number

The value read is 23 the state of the trace table is as shown in Figure 9.12. The loop condition is tested and is true because the content of NUMBER is 23 and not 999. The loop body is executed again. The statement:

PosSum = PosSum + Number

is executed. The value of PosSum becomes 23 and the state of the trace table is as shown in Figure 8.13. The other tests are ignored because they are false. The next instruction to be executed is

read Number

The number read is 30. The body of the loop is executed again because the condition (Number <> 999) is true. The instruction

PosSum = PosSum + Number

is executed. The value of PosSum become, 53 (23 + 30) and the state of the trace table is as shown in Figure 8.14. The other tests are ignored because they are false. The next instruction to be executed is:

read Number

NCount	ZCount	PosSum	Number
0	0	0	-12
1			23

Figure 8.12

NCount	ZCount	PosSum	Number
0	0	0	-12
1		23	23

Figure 8.13

The number read is 0. The body of the loop is executed again because the condition (Number <> 999) is true. The instruction:

ZCount = ZCount + 1

is executed. The value of ZCount become 1 (0 + 1) and the state of the trace table is as shown in Figure 8.15. The other tests are ignored because they are false. The next instruction to be executed is:

read Number

The processing of the loop block continues until the number 999 is read. At this point the state of the trace table is as shown in Figure 9.15. The loop is terminated and the print statements are executed. Since the final information in the trace table is correct, one can therefore conclude, that the logic of the program design is sound.

NCount	ZCount	PosSum	Number
0	0	0	-12
1		23	23
		53	30

Figure 8.14

NCount	ZCount	PosSum	Number
0	0	0	-12
1	1	23	23
2	2	53	30
	3	103	0
		118	-120
			50
			0
			0
			15
			999

Figure 8.15

Review Question

1. Do a trace table to show the final state of the variable in the following pseudocode:

```
Sum = 0
N = 10
while N < 40 do
        Sum = Sum + N
        print N, Sum
        N = N + 5
endwhile
```

2. Use a trace table to determine what is printed by the following algorithm when n = 5?

```
if (n = 1) or (n = 2) then
        h = 1
else
        f = 1
        g = 1
        for j = 1 to n -1 do
                h = f + g
                f = g
                g = h
                print h
        endfor
endif
print f, g
stop
```

3 Use a trace table to determine what is printed by the following algorithm.

```
Sum = 0
N = 20
while N < 30 do
        Sum = Sum + N
        print N, Sum
        N = N + 3
endwhile
```

4. Use a trace table to determine what is printed by the following algorithm.

```
Count = 1
X = 2
while Count < 25 do
        X = X + 2
        print Count, X
        Count = Count + 5
endwhile
```

5. Complete the trace table below for the following algorithm given that the number 6 is the input value:

```
read X
for M = 1 to X do
        Y = X - M
        Z = 5*Y - M
endfor
print Z
```

X	M	Y	Z
6	1	5	
6			
6		3	
6			
6			
6			-6

6. Use a trace table to determine what is printed by the following algorithm.

```
X = 5
K = 10
Sum = 45

while Sum < 75 do
        Sum = Sum + K
        print K
        K = K + X
endwhile
print Sum
```

7.

```
Difference = 0
input A, B
if A <= B then
        Difference = B - A
else
        Difference = B - A
endif

print Difference
```

What is printed by the algorithm above if the input values are the following?

(i) 20 30
(ii) 100 100
(iii) 50 50