Lesson

8

TRACE TABLE

Trace table

A trace table is an important tool for testing the logics of a pseudocode for correctness. A trace table is a rectangular array of rows and columns. The column headings are the variables in the pseudocode. As the instructions in the pseudocode are carried out and the variables are modified, the changes are recorded in the appropriate column in the table. When the pseudocode terminates, the final values in the trace table should reflect the correct result.

QUESTION **Test the correctness of the following pseudocode**

```
sum = 0
read number
While number <> 0 do
      sum = sum + number
      read number
endwhile

print "Sum of numbers is", sum
```

using a trace table. Use the following 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 statement in the pseudocode is:

```
SUM = 0
```

When this statement is carried out the trace tables is as shown in **FIGURE 8-1.** The second statement in the pseudocode is:

```
read NUMBER
```

The first number in the sequence of inputs is 12. When this statement is carried out the trace table is as shown in **FIGURE 8- 2**. The condition statement:

```
NUMBER <> 0
```

is tested. We enter the loop because the content of **NUMBER** is 12 {***this is the reason why we had to read our first number outside the loop***}. The first statement 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 carried out is :

```
read NUMBER
```

After this statement is carried out **the state of the trace table is as shown in FIGURE 8-4** because the next value in the sequence of inputs is 23.

We are now in the loop. We therefore continue in the loop until the termination condition

```
NUMBER <> 0
```

becomes false. Again, the loop condition is tested and the next statement to be carried out is:

```
SUM = SUM + NUMBER
```

because the content of NUMBER is 23 and not 0.

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

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 carried out is:

```
read NUMBER
```

The new state of the trace table is as shown in **FIGURE 8-6.** The next statement to be carried out is:

```
SUM = SUM + NUMBER
```

because the content of NUMBER is still not 0. The state of the trace table is as shown in **FIGURE 8-7**. The next statement to be carried out is:

```
read NUMBER
```

The number read is 0. The loop terminates because the content of NUMBER is now 0 and the conditional statement

```
NUMBER <> 0
```

becomes false. The loop is exited and the next statement to be carried out is :

```
print "Sum of numbers is", SUM
```

This statement produces the following output:

```
Sum of numbers is 69
```

and the pseudocode comes to an end. The final state of the trace table is as shown in **FIGURE 8-8**. The final content of sum being 69 is the correct result. The logics of the pseudocode is therefore sound.

NCOUNT	ZCOUNT	POSSUM	NUMBER

FIGURE 8-9

PSEUDOCODE

```
NCOUNT = 0
ZCOUNT = 0
POSSUM = 0

read NUMBER

while NUMBER <> 999 do

        if  NUMBER < 0 then
                NCOUNT = NCOUNT + 1
        endif

        if NUMBER = 0 then
                ZCOUNT = ZCOUNT + 1
        endif

        if NUMBER > 0 the
                POSSUM = POSSUM + NUMBER
        endif
        read NUMBER
endwhile

print "Number of negative numbers is", NCOUNT
print "Number of zeros is", ZCOUNT
print "Sum of positive numbers is", POSSUM
```

Assignment statement to initialize variable

If statement to test if number is less than 0

If statement to test if number is 0

If statement to test if number is positive

Statement to print content of variables.

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

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

SOLUTION Figure 8-9 shows the layout of the trace table for the pseudocode. The four variables used in the pseudocode(**NCOUNT, ZCOUNT, POSSUM , NUMBER**) form 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 pseudocode are:

```
NCOUNT = 0
ZCOUNT = 0
POSSUM = 0

read NUMBER
```

After the four statements above are executed, the state of the trace 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 number is carried out, the content of NUMBER is -12 . The test is true therefore the statement:

```
NCOUNT = NCOUNT + 1
```

is carried out. 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 carried out is:

```
read NUMBER
```

The value read is 23.The state of the trace table is as shown in **FIGURE 8-12**. The loop condition is tested and is true because the content of NUMBER is 23 and not 999. The test on the number is carried out and the third test is true therefore the statement:

```
POSSUM = POSSUM + NUMBER
```

is carried out. The state of the trace table is as shown in **FIGURE 8-13**. The other tests are ignored. The next statement to be carried out is:

```
read NUMBER
```

The number read is 30 , the body of the loop is repeated. The third test is selected and the statement:

```
POSSUM = POSSUM + NUMBER  (23 + 30)
```

The state of the trace table is shown in **FIGURE 8-14**. The next statement to be carried out is:

```
read NUMBER
```

The number read is 0. The body of the loop is repeated and the second test is selected and the statement:

```
ZCOUNT = ZCOUNT + 1
```

The next number is read and the processing of the loop body continues until the number 999 is read . At this point the state of the trace table is as shown in **FIGURE 8-15**. 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 logics are sound.

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

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