Stepwise Refinement Example

Jacek Wasilewski

Problem

- Given a positive integer, N, define the 3N+1 sequence starting from N as follows:
  - if N is an even number, then divide N by two;
  - if N is odd, then multiply N by 3 and add 1.
- Continue to generate numbers in this way until N becomes equal to 1.
- Write a program that will take a positive integer and will print out the 3N+1 sequence starting from that integer.
- The program should also count and print out the number of terms in the sequence.

Example

- For example:

N = 3 (odd)

N = 3 * 3 + 1 = 10 (even)

N = 10 / 2 = 5 (odd)

N = 5 * 3 + 1 = 16 (even)

N = 8 / 2 = 4 (even)

N = 4 / 2 = 2 (even)

N = 2 / 2 = 1 (stop)

Code (1)

```
Read a positive integer N;
Compute, print, and count each number in the sequence;
Output the number of terms;
```

Code (2)

```
Read a positive integer N;
while N is not 1:
    Compute N = next term;
    Output N;
    Count this term;
Output the number of terms;
```

Code (3)

```
Read a positive integer N;
while N is not 1:
    if N is even:
        Compute N = N / 2;
    else:
        Compute N = 3 * N + 1;
    Output N;
    Count this term;
Output the number of terms;
```

Code (4)

```
Read a positive integer N;
Let counter  = 0;
while N is not 1:
    if N is even:
        Compute N = N / 2;
    else:
        Compute N = 3 * N + 1;
    Output N;
    Add 1 to counter;
Output the counter;
```

Code (5)

```
Read N;
if N is not positive:
    Stop the program;
Let counter  = 0;
while N is not 1:
    if N is even:
        Compute N = N / 2;
    else:
        Compute N = 3 * N + 1;
    Output N;
    Add 1 to counter;
Output the counter;
```