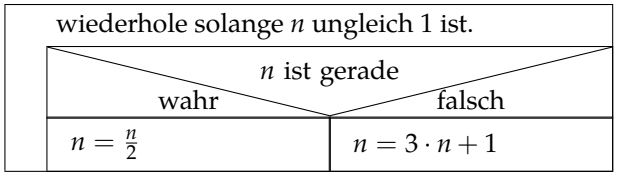
Abitur 2019 IV

Das Collatz-Problem ist ein immer noch ungelöstes Problem der Mathematik. Dabei geht es um Zahlenfolgen, die nach folgendem Algorithmus gebildet werden, wobei der Eingabewert n eine natürliche Zahl größer 0 ist:

collatzfolge(n)



Obwohl der Algorithmus sehr einfach ist, ist bis heute ungeklärt, ob er tatsächlich bei jedem beliebigen Startwert von n nach endlich vielen Durchläufen der Wiederholung terminiert.

(a) Geben Sie die Zahlenfolge an, die man mit dem Startwert 7 erhält, wenn n nach jedem Durchlauf der Wiederholung ausgegeben wird.

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

(b) Beschreiben Sie, wie man mithilfe der ganzzahligen Division ohne Rest prüfen kann, ob eine Zahl a durch eine andere Zahl b teilbar ist.

Wenn man das Ergebnis der Division der beiden Zahlen a und b mit b multipliziert und nach der Mulitplikation als Ergebnis wieder die Zahl a feststeht, dann handelt es sich um eine Division ohne Rest, ergibt sich eine Zahl, die kleiner als a ist, so handelt es sich um eine Division mit Rest.

(c) Geben Sie ein Programm für die Registermaschine an, das den gegebenen Algorithmus `collatzfolge(n)` umsetzt, wobei zusätzlich die Anzahl der Durchläufe der Wiederholung bestimmt werden soll. Der Startwert für n steht am Anfang bereits in Speicherzelle 100.

Ohne Modulo

```
# n:      100
# anzahl: 101
                LOADI 0
                STORE 101

# WHILE n <> 1
solange:        LOAD 100
                SUBI 1
                JMPZ ende

# anzahl := anzahl + 1;
zaehler:        LOAD 101
                ADDI 1
                STORE 101

# Poorman's Modulo
# IF (n % 2) = 0 THEN
modulo:         LOAD 100
                DIVI 2
                MULI 2
                SUB 100
                JMPN ist_ungerade

# n := n / 2;
ist_gerade:     LOAD 100
                DIVI 2
                STORE 100
                JMP solange

# n := 3 * n +  1;
ist_ungerade:   LOAD 100
                MULI 3
                ADDI 1
                STORE 100
                JMP solange

ende:           HOLD
```

Mit Modulo

```
# n:      100
# anzahl: 101
                LOADI 0
                STORE 101

# WHILE n <> 1
solange:        LOAD 100
                CMPI 1
                JMPZ ende

# IF (n % 2) = 0 THEN
bedingung:      LOAD 100
                MODI 2
                JMPNZ ist_ungerade

# n := n / 2;
ist_gerade:     LOAD 100
                DIVI 2
                STORE 100
                JMP zaehler

# n := 3 * n +  1;
ist_ungerade:   LOADI 3
                MUL 100
                ADDI 1
                STORE 100

# anzahl := anzahl + 1;
zaehler:        LOAD 101
                ADDI 1
                STORE 101
                JMP solange

ende:           HOLD
```

Minisprache

```
PROGRAM collatz;
VAR n, anzahl;

BEGIN
  n := 7;
  anzahl := 0;
  WHILE n <> 1 DO
    IF (n % 2) = 0 THEN
      n := n / 2;
    ELSE
      n := 3 * n +  1;
    END;
    anzahl := anzahl + 1;
  END
END collatz.
```

Java

```
public class Collatz {
  public static void main(String[] args) {
    int n = 7;
    int anzahl = 0;
    while (n != 1) {
      if (n % 2 == 0) {
        n = n / 2;
      } else {
        n = 3 * n + 1;
      }
      anzahl++;
      System.out.println(n);
    }
    System.out.println("Anzahl an Durchläufen " + anzahl);
  }
}
```