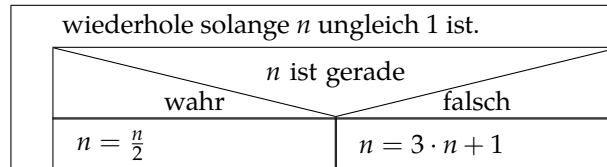


Abitur 2019 IV

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 Multiplikation 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.

Mit Modulo

```
1  # n:      100
2  # anzahl: 101
3          LOADI 0
4          STORE 101
5
6  # WHILE n <> 1
7  solange:  LOAD 100
8            CMPI 1
9            JMPZ ende
10
11 # IF (n % 2) = 0 THEN
12 bedingung: LOAD 100
13            MODI 2
14            JMPNZ ist_ungerade
```

```

15
16 # n := n / 2;
17 ist_gerade:  LOAD 100
18             DIVI 2
19             STORE 100
20             JMP zaehler
21
22 # n := 3 * n + 1;
23 ist_ungerade: LOADI 3
24             MUL 100
25             ADDI 1
26             STORE 100
27
28 # anzahl := anzahl + 1;
29 zaehler:    LOAD 101
30             ADDI 1
31             STORE 101
32             JMP solange
33
34 ende:      HOLD

```

Ohne Modulo

```

1             LOADI 0
2             STORE 101
3
4 solange:    LOAD 100
5             SUBI 1
6             JMPZ ende
7
8 zaehler:    LOAD 101
9             ADDI 1
10            STORE 101
11
12 modulo:    LOAD 100
13            DIVI 2
14            MULI 2
15            SUB 100
16            JMPN ist_ungerade
17
18 ist_gerade: LOAD 100
19            DIVI 2
20            STORE 100
21            JMP solange
22
23 ist_ungerade: LOAD 100
24            MULI 3
25            ADDI 1
26            STORE 100
27            JMP solange
28 ende:      HOLD

```

Minisprache

```

1 PROGRAM collatz;
2 VAR n, anzahl;
3
4 BEGIN
5     n := 7;
6     anzahl := 0;

```

```

7  WHILE n <> 1 DO
8      IF (n % 2) = 0 THEN
9          n := n / 2;
10     ELSE
11         n := 3 * n + 1;
12     END;
13     anzahl := anzahl + 1;
14 END
15 END collatz.

1  package org.bsclangaul.aufgaben.tech_info.assembler;
2
3  public class Collatz {
4      public static void main(String[] args) {
5          int n = 7;
6          int anzahl = 0;
7          while (n != 1) {
8              if (n % 2 == 0) {
9                  n = n / 2;
10             } else {
11                 n = 3 * n + 1;
12             }
13             anzahl++;
14             System.out.println(n);
15         }
16         System.out.println("Anzahl an Durchläufen " + anzahl);
17     }
18 }

```