

# CPU-Rollenspiel - Programm „Fibonacci-Zahlen“

## Ausgangslage

Das **Programm** liegt im Speicher ab Speicherzelle 0 (Anfang)

Die **Daten** liegen im Speicher ab Speicherzelle 100:

Speicherplatz mit Adresse 100 enthält die Anzahl der geplanten Durchläufe (z.B. die Zahl 3)

Speicherplatz mit Adresse 101 enthält die Zahl 1

Speicherplatz mit Adresse 102 enthält die Zahl 1

## Programm (Assemblersprache)

```
# Anzahl der geplanten Durchläufe aus Speicherzelle 100 holen
```

```
LOAD (100) ->D
```

```
#Sprungmarke (Speicheradresse merken)
```

```
loop:
```

```
# Die beiden vorherigen F-Zahlen aus dem Speicher (Zellen 101 und 102) holen
```

```
LOAD (101) -> A
```

```
LOAD (102) -> B
```

```
# Addieren der Register A und B, Resultat im Register C speichern
```

```
ADD A,B -> C
```

```
# Erste Fibonacci-Zahl mit der zweiten überschreiben
```

```
LOAD (102) -> A
```

```
LOAD A -> (101)
```

```
# register C als zweite Zahl speichern
```

```
LOAD C-> (102)
```

```
# Anzahl verbleibende Durchläufe um 1 reduzieren („dekrementieren“)
```

```
DEC D
```

```
# wenn register D noch nicht gleich null -> zur Sprungmarke „loop“ springen
```

```
JMPDNZ loop:
```

```
# Resultat ausgeben (z.B. auf Display)
```

```
OUT (102)
```

# Programm (Maschinensprache)

| Adresse (Bin) | Adresse (Dec) | Inhalt (Bin) | Inhalt (Dec) |
|---------------|---------------|--------------|--------------|
| 0000 0000     | 0             | 0000 0010    | 2            |
| 0000 0001     | 1             | 0110 0100    | 100          |
| 0000 0010     | 2             | 0000 0000    | 0            |
| 0000 0011     | 3             | 0110 0101    | 101          |
| 0000 0100     | 4             | 0000 0001    | 1            |
| 0000 0101     | 5             | 0110 0110    | 102          |
| 0000 0110     | 6             | 0000 0005    | 5            |
| 0000 0111     | 7             | 0000 0000    | 0            |
| 0000 1000     | 8             | 0110 0110    | 102          |
| 0000 1001     | 9             | 0000 0011    | 3            |
| 0000 1010     | 10            | 0110 0101    | 101          |
| 0000 1011     | 11            | 0000 0100    | 4            |
| 0000 1100     | 12            | 0110 0110    | 102          |
| 0000 1101     | 13            | 0000 0110    | 6            |
| 0000 1110     | 14            | 0000 0111    | 7            |
| 0000 1111     | 15            | 0000 0010    | 2            |
| 0001 0000     | 16            | 0000 1000    | 8            |
| 0001 0001     | 17            | 0110 0110    | 102          |

## Daten (Anfangszustand)

|           |     |           |   |                                               |
|-----------|-----|-----------|---|-----------------------------------------------|
| 0110 0100 | 100 | 0000 0011 | 3 | Erste Fibonacci-Zahl<br>Zweite Fibonacci-Zahl |
| 0110 0101 | 101 | 0000 0001 | 1 |                                               |
| 0110 0110 | 102 | 0000 0001 | 1 |                                               |