

## Prozedurale Programmierung (PPR)

WS 2020/21

### Praktikumsaufgabe 1 (ULAM-Folge)

Das *Collatz-Problem* gehört zu den ungelösten Problemen der Mathematik. Man beginne mit einer beliebigen natürlichen Zahl  $a_n$  und bilde damit die rekursive Zahlenfolge

$$a_{n+1} = \begin{cases} \frac{1}{2}a_n & \text{falls } a_n > 1 \text{ und } a_n \text{ gerade} \\ 3a_n + 1 & \text{falls } a_n > 0 \text{ und } a_n \text{ ungerade} \end{cases}$$

Die Folge endet, wenn sie den Wert 1 erreicht, d.h. für  $a_n = 1$  ist  $a_{n+1}$  nicht mehr definiert. Beispielsweise erhält man für

- $a_0 = 5$  die Folge 5, 16, 8, 4, 2, 1.
- $a_0 = 7$  die Folge 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Die Vermutung, dass für jede natürliche Zahl  $a_0$  die Folge nach endlich vielen Schritten den Wert 1 erreicht, konnte bisher weder bewiesen noch widerlegt werden. Computer haben alle Zahlen bis  $3 \cdot 2^{53}$  (Stand 1999) durchprobiert; immer endet die Zahlenfolge mit 1, was die Vermutung bestätigt oder zumindest nicht widerlegt.

Die im Collatz-Problem verwendete Funktion wird manchmal auch *ULAM-Funktion* genannt.

- Schreiben Sie eine C-Funktion

```
int ulam_max(int a0);
```

die für eine ganze positive Zahl  $a_0$  den maximalen Wert in der Folge ihrer ULAM-Werte berechnet. Hat  $a_0$  einen Wert kleiner oder gleich 0, soll die Funktion  $-1$  liefern. Beispielsweise sollen die Aufrufe

- `ulam_max(5)` den Wert 16 und
- `ulam_max(7)` den Wert 52

liefern.

- Schreiben Sie eine C-Funktion

```
int ulam_twins(int limit);
```

die prüft, ob es im Intervall von 1 bis einschließlich `limit` zwei benachbarte Werte  $a_0$  und  $a_0 + 1$  gibt, für die `ulam_max(a0) = ulam_max(a0 + 1)` gilt. Ein solches Paar wird als *ULAM-Zwilling* bezeichnet.

Die Funktion soll  $a_0$  liefern, wenn ein Paar gefunden wurde und  $a_0$  der kleinere Wert in einem solchen Paar ist. Sind mehrere Paare im Intervall enthalten, soll das letzte Paar gesucht werden. Die Funktion soll  $-1$  zurückgeben, wenn kein solches Zwillingsspaar gefunden wurde. Beispielsweise soll die Aufrufe

- `ulam_twins(6)` den Wert 5 liefern, da sowohl `ulam_max(5)` als auch `ulam_max(6)` den Wert 16 ergeben und
- `ulam_twins(5)` den Wert -1 liefern, da 6 nicht im Intervall  $[1, 5]$  enthalten ist.

- Schreiben Sie eine C-Funktion

```
int ulam_multiples(int limit, int number);
```

die prüft, ob im Intervall von 1 bis einschließlich `limit` *ULAM-Mehrlinge* mit der Anzahl `number` vollständig enthalten sind, d.h. für `number = 3` sollen Drillinge, für `number = 4` Vierlinge usw. gesucht werden.

Die Funktion soll `a0` liefern, wenn Mehrlinge gefunden wurden und `a0` der kleinste Wert ist, der zu den Mehrlingen gehört. Sind weitere Mehrlingsgruppen vollständig im Intervall enthalten, soll der kleinste Wert der letzten Gruppe zurück gegeben werden. Die Funktion soll -1 zurückgeben, wenn die Parameter nicht sinnvoll sind oder keine solchen Mehrlinge gefunden wurden. Beispielsweise sollen die Aufrufe

- `ulam_multiples( 10, 2)` den Wert 5
- `ulam_multiples(1000, 3)` den Wert 972
- `ulam_multiples( 108, 3)` den Wert -1
- `ulam_multiples( 391, 6)` den Wert 386

liefern.

### Hinweise zu dieser Praktikumsaufgabe

1. Für dieses Blatt sind zunächst nur die Kapitel 4, 5, 7 und 10 der PPR-Programmierrichtlinien relevant.