

Aufgabe 6.1

Schreibe ein Programm `count1.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- mit einer `for`-Schleife und der Zählvariablen `i` die ersten `n` natürlichen Zahlen ausgibt.

Aufgabe 6.2

Schreibe ein Programm `count2.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- mit einer `for`-Schleife und der Zählvariablen `i` die ersten `n` *ungeraden* natürlichen Zahlen ausgibt.

Aufgabe 6.3

Schreibe ein Programm `squares.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- mit einer `for`-Schleife und der Variablen `i` die Quadrate der ersten `n` natürlichen Zahlen ausgibt.

Aufgabe 6.4

Schreibe ein Programm `sum1.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- die Variablen `summe` und `i` jeweils mit dem Wert 0 initialisiert;
- mit einer `for`-Schleife und den Variablen `summe` und `i` die Summe der ersten `n` natürlichen Zahlen berechnet;
- den Wert der Summe ausgibt.

Aufgabe 6.5

Schreibe ein Programm `sum2.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- die Variable `summe` und `i` jeweils mit dem Wert 0 initialisiert;
- mit einer for-Schleife und den Variablen `summe` und `i` die Summe der ersten `n ungeraden` natürlichen Zahlen berechnet;
- den Wert der Summe ausgibt.

Aufgabe 6.6

Schreibe ein Programm `stars.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in ein ganze Zahl transformiert und in der Variablen `n` speichert;
- mit einer for-Schleife eine Figur mit `n` Zeilen zeichnet, in der die `i`-te Zeile aus `i` Sternen besteht;

Hinweis: eine Folge von `i` Sternen lässt sich direkt durch die Stringmultiplikation `i * '*'` erzeugen.

Aufgabe 6.7

Die Collatz-Folge wird wie folgt definiert:

- Wähle eine natürliche Zahl $n > 1$,
- Wiederhole, bis $n = 1$:
 - falls n gerade ist, berechne $n = n/2$
 - sonst berechne $n = 3n + 1$

Beispiel: 6, 3, 10, 5, 16, 8, 4, 2, 1

Schreibe ein Programm `collatz.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- mit einer while-Schleife die Glieder der Collatz-Folge berechnet und ausgibt, bis der Wert $n = 1$ erreicht wird.

Aufgabe 6.8

Schreibe ein Programm `primzahltest.py`, das

- dazu auffordert, eine natürliche Zahl einzugeben, die Eingabe in eine ganze Zahl umwandelt und in der Variablen `n` speichert;
- eine Variable `isPrime` mit dem Wert `True` anlegt;
- in einer `for`-Schleife für die Variable `i` von 2 bis und mit `n-1` prüft, ob `n` ohne Rest durch `i` teilbar ist; falls ja, wird der Variable `isPrime` der Wert `False` zugewiesen und die Schleife mit dem Schlüsselwort `break` abgebrochen;
- das Ergebnis des Tests (Primzahl oder nicht) ausgibt.