Übungsblatt 3

02.05. und 05.05

Problem 3.1: Collatz-Folge (`while`-Schleife, Fallunterscheidung)

Eine *Collatz-Folge* startet mit einer positiven ganzen Zahl; der Nachfolger eines Folgenglieds n ist $n/2$, falls n gerade ist und $3n+1$, falls n ungerade ist.

Schreiben Sie ein Programm, das einen Startwert von `cin` einliest und so lange Folgenglieder berechnet und ausdruckt, bis zum ersten Mal die 1 erreicht wird. (Man vermutet, dass für beliebige Startwerte irgendwann die 1 erreicht wird; danach ist der Rest der Folge langweilig: $1, 4, 2, 1, \ldots$)

Tipp: Zum Prüfen, ob eine Zahl gerade ist, den Modulo-Operator % verwenden.

Ein Beispiellauf für das Programm (Eingabe unterstrichen):

```
Startwert: 3
10
5
16
8
4
2
1
```

Problem 3.2: Polygonfläche (Schleifen)

Zunächst noch eine Aufgabe zu Schleifen.

Es seien $(x_1, y_1), \ldots, (x_n, y_n)$ Eckpunkte eines einfachen Polygons (*einfach* heißt hier: Die Kanten des Polygons überschneiden sich nicht, abgesehen davon, dass je zwei benachbarte Kanten einen Eckpunkt gemeinsam haben).

Die orientierte Fläche A des Polygons (positiv, falls der Rand des Polygons im mathematisch positiven Sinn durchlaufen wird, sonst negativ) lässt sich nach der *Gaußschen Trapezformel* berechnen als

$$A = \frac{1}{2}\left((y_n + y_1)\cdot(x_n - x_1) + \sum_{i=2}^{n}(y_{i-1} + y_i)\cdot(x_{i-1} - x_i)\right)$$

1. Zunächst betrachten wir ein Dreieck, also $n = 3$.

   Schreiben Sie ein Programm, das die drei Koordinatenpaare (sechs Variablen vom Typ `double`) von `cin` einliest und dann die Dreiecksfläche berechnet und ausgibt.

   Tipp: Hier ist noch keine Schleife notwendig – die Formel zur Flächenberechnung lässt sich ja für festes n leicht als Ausdruck ohne Summenzeichen schreiben.

2. Nun soll der Benutzer die Zahl n der Eckpunkte eingeben können und anschließend entsprechend viele Koordinatenpaare.

   Für dieses Programm ist die Beobachtung wichtig, dass wir nicht erst alle Koordinaten einlesen und speichern müssen (wie man das machen würde, lernen wir noch), sondern dass es für das nächste Summenglied reicht, die beiden Endpunkte (x_{i-1}, y_{i-1}) und (x_i, y_i) *der zuletzt eingefügten Kante* zu kennen – im Übrigen müssen wir nur noch den ganz zum Schluss wieder benötigten Anfangspunkt (x_1, y_1) speichern.

Vorschlag für den Aufbau Ihres Programms:

- Lesen Sie die Anzahl n der Ecken ein und die ersten beiden Punkte wie in Teil 1.
- Initialisieren Sie die Summe mit dem Wert $(y_1 + y_2) \cdot (x_1 - x_2)$
- Beginnen Sie dann eine Schleife $i = 3, \ldots, n$:
  - Koordinaten x_i und y_i einlesen.
  - Zur bisherigen Summe den Term $(y_{i-1} + y_i) \cdot (x_{i-1} - x_i)$ addieren.
  - Aus der Summe und dem Wert $(y_i + y_1) \cdot (x_i - x_1)$ können Sie in jedem Schleifendurchlauf die Fläche des bisherigen Polygons berechnen und zur Kontrolle ausgeben.
  - Bevor es zum nächsten Punkt weitergeht, müssen Sie dann noch x_i und y_i geeignet umspeichern, so dass sie im nächsten Durchlauf $(i \rightsquigarrow i + 1)$ als x_{i-1} und y_{i-1} zur Verfügung stehen.

Dieses Programm ist schon komplizierter – überlegen Sie sich vorher, welche Variablen Sie für was verwenden wollen, probieren Sie Ihren Algorithmus erstmal mit Papier und Bleistift aus und tasten Sie sich beim Ausprobieren langsam vom Dreieck über ein Viereck zu komplizierteren Polygonen voran!