

Übung 7

Ziel der Übung:

- Verwendung von Arrays (Definition, Speichern von Daten im Array, Zugriff auf Arrayelemente)
- Einsatz von Arrays in Programmen
- Wiederholung: verschachtelte Schleifen

Aufgabe 1: Auswertung von Messdaten

In einer Lebensmittel-Produktionsanlage werden Packungen verschiedener Müslisorten mit jeweils 250 Gramm abgefüllt. Bedingt durch die Zusammensetzung kommt es dabei zu leicht unterschiedlichen Gewichten der Packungen.

Schreiben Sie ein C++-Programm, das für $n = 10$ Packungen das Gewicht x nacheinander einliest und das arithmetische Mittel \bar{x} sowie den mittleren absoluten Fehler nach den folgenden Formeln berechnet und ausgibt.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \qquad fehler = \frac{1}{n} \sum_{i=1}^n |sollwert - x_i|$$

Verwenden Sie ein **Array** zur Speicherung der Messwerte.

Hinweis: Die Eingaben sowie das Speichern der Gewichte in das Array kann über eine Schleife erledigt werden. Die Funktion `abs()` liefert den absoluten Wert einer Zahl.

Beispiel: Die Aus- und Eingaben Ihres Programms sollen wie folgt aussehen (die Ausgaben sind schwarz und die Eingaben grau dargestellt):

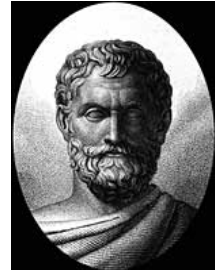
```
Gewicht von Packung 1: 251
Gewicht von Packung 2: 245
Gewicht von Packung 3: 253
Gewicht von Packung 4: 252
Gewicht von Packung 5: 251
Gewicht von Packung 6: 250
Gewicht von Packung 7: 246
Gewicht von Packung 8: 249
Gewicht von Packung 9: 252
Gewicht von Packung 10: 257
```

```
Mittelwert: 250.6
Fehler:      2.6
```

Aufgabe 2: Primzahlen ermitteln: Sieb des Eratosthenes

Eine einfache Idee zur Ermittlung von Primzahlen ist folgende (*Sieb des Eratosthenes*, 240 v.Chr.):

Eine Liste aller Zahlen bis n wird angelegt. Alle Zahlen, die das Produkt der Multiplikation zweier Zahlen zwischen 2 und n sind, werden aus der Liste gestrichen.



Beispiel:

Ausgangszustand des Arrays

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

Nach dem Löschen der Zahlen, die das Produkt zweier Zahlen sind, verbleiben die 0 und 1 und nachfolgend die Primzahlen im Feld:

0	1	2	3	0	5	0	7	0	0	0	11	0	13	0	0	0	17
---	---	---	---	---	---	---	---	---	---	---	----	---	----	---	---	---	----

Schreiben Sie ein **C++-Programm** basierend auf dieser Idee:

1. Entwickeln Sie zunächst den **Algorithmus**.
2. Legen Sie in Ihrem Programm ein **Feld** zur Speicherung von 10000 Zahlen fest.
3. Speichern Sie darin die natürlichen **Zahlen von 0 bis 9999**.
4. **Löschen** Sie aus dem Feld alle Zahlen, die sich als **Produkt der Multiplikation** zweier Zahlen zwischen 2 und n ergeben. (Löschen bedeutet hier auf 0 setzen)
5. **Geben** Sie die **verbleibenden Primzahlen** aus.

Hausaufgaben Serie 7

Hausaufgaben mit Namen, Studiengang und Matrikelnummer unter „**Aufgaben**“ auf StudIP hochladen.
Abgabe bis **07.12.2021** (ÜG-1), **08.12.2021** (ÜG-4), **09.12.2021** (ÜG-2) und **10.12.2021** (ÜG-3).

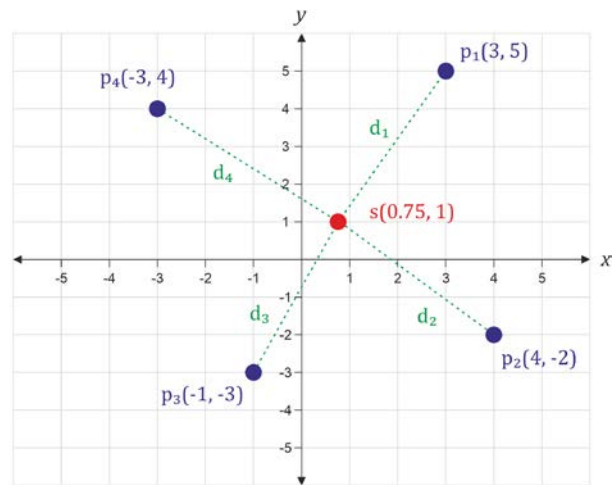
Aufgabe 1: Punktkoordinaten im Array speichern und verwenden

Schreiben Sie ein **C++-Programm**, das $n = 4$ paarweise Punkte p_1, \dots, p_n aus dem reellen Vektorraum einliest. Speichern Sie die Punktkoordinaten $p = \{x, y\}$ in ein zweidimensionales Array `punkte[n][2]`. Berechnen Sie zunächst den einfachen **geometrischen Schwerpunkt** s der gespeicherten Punkte. Die Koordinaten des geometrischen Schwerpunkts $s = (x_s, y_s)$ sind wie folgt definiert:

$$x_s = \frac{1}{n} \sum_{i=0}^{n-1} x_i \text{ und } y_s = \frac{1}{n} \sum_{i=0}^{n-1} y_i$$

Berechnen Sie anschließend den **euklidischen Abstand** aller Punkte p_1, \dots, p_n vom Schwerpunkt s . Der euklidische Abstand d_i eines Punktes $p_i = \{x_i, y_i\}$ vom Schwerpunkt $s = (x_s, y_s)$ ist wie folgt definiert:

$$d_i = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2}$$



Geben Sie den geometrischen Schwerpunkt sowie den euklidischen Abstand aller Punkte zum Schwerpunkt aus. Verwenden Sie für das Einlesen und Speichern der Koordinaten sowie für die Berechnungen jeweils eine **Schleife**.

Beispiel: Die Aus- und Eingaben Ihres Programms sollen wie folgt aussehen (die Ausgaben sind schwarz und die Eingaben grau dargestellt):

Punkt 1:

x = 3

y = 5

Punkt 2:

x = 4

y = -2

Punkt 3:

x = -1

y = -3

Punkt 4:

x = -3

y = 4

Geometrischer Schwerpunkt: 0.75, 1

Distanz zum Punkt 1: 4.59

Distanz zum Punkt 2: 4.42

Distanz zum Punkt 3: 4.37

Distanz zum Punkt 4: 4.80

(13 Punkte)

Aufgabe 2: Funktion zum Verdoppeln der geraden Elemente eines Arrays

Die Übergabe eines Arrays als Eingabeparameter an eine Funktion wird intern so realisiert, dass die Adresse des Feldes übergeben wird. Man hat hier also das gleiche Verhalten wie bei einem "call-by-reference"-Aufruf.

Teil a) Schreiben Sie eine C++-Funktion `int verdoppeln(int arr[], int n)`, die die **Werte der geraden Elemente** eines Arrays **verdoppelt** und die **die Summen aller Elemente nach der Verdoppelung** zurückliefert.

Teil b) Schreiben Sie ein **Hauptprogramm**, in dem ein Array vom Datentyp `int` mit sinnvollen Werten initialisiert wird, d.h. ohne Eingabe durch den Nutzer. Geben Sie dieses Array aus. Übergeben Sie das Array und dessen Länge an die Funktion `verdoppeln` und geben Sie anschließend die Elemente des Arrays sowie die durch die Funktion berechnete Summe aus.

Beispiel:

Array vorher: 2 13 6 5 7 24 32 41 48 39

Array nachher: 4 13 12 5 7 48 64 41 96 39

Summe: 329

(10 Punkte)