

## Übung 4

### Ziel dieser Übung und Hausaufgaben:

- Verschiedene Varianten von Schleifen in C++-Programmen üben
- algorithmisches Denken festigen

### Aufgabe 1: Primzahlberechnung

Schreiben Sie bitte ein **C++-Programm**, das testet, ob eine eingelesene Zahl eine **Primzahl** ist oder nicht.

**Tipp:** Zur Prüfung, ob eine Zahl  $p$  eine Primzahl ist, bietet es sich an, zu testen, ob  $p$  nur durch 1 und  $p$ , jedoch **nicht durch eine andere Zahl teilbar** ist. Dazu kann man **für alle Zahlen von  $2 \dots p - 1$  nacheinander testen**, ob diese ein Teiler der Zahl  $p$  sind.

**Beispiel:** Die Ausgaben sind schwarz und die Eingabe grau dargestellt.

Bitte geben Sie eine Zahl ein: 5

Ergebnis: Primzahl

Erstellen Sie ein **C++-Programm** für diese Aufgabe. Wenn Ihnen ein Optimierungsschritt einfällt, können Sie diesen gern in das Programm aufnehmen.

### Aufgabe 2: Ausgabe aller Primzahlen

In dieser Aufgabe sollen Sie das Programm aus *Aufgabe 1* erweitern. Schreiben Sie ein **C++-Programm**, das für die Zahlen bis zu einer einzulesenden Grenze berechnet, ob diese **Primzahlen sind oder nicht**. **Alle Primzahlen** sollen **ausgegeben** werden.

**Beispiel:** Die Ausgaben sind schwarz und die Eingabe grau dargestellt.

Bis zu welcher Zahl sollen die Primzahlen berechnet werden: 15

2 3 5 7 11 13

### Zusatzaufgabe: Optimierung der Primzahlberechnung

Fallen Ihnen Möglichkeiten ein, wie Sie Ihr Programm effizienter schreiben können, wie Sie die Anzahl der Berechnungsschritte im Vergleich zu Ihrer ersten Lösung in *Aufgabe 2* verringern können?

## Hausaufgaben Serie 4

Hausaufgaben mit Namen, Studiengang und Matrikelnummer unter „**Aufgaben**“ auf StudIP hochladen.  
Abgabe bis **16.11.2021** (ÜG-1), **17.11.2021** (ÜG-4), **18.11.2021** (ÜG-2) und **19.11.2021** (ÜG-3).

### Aufgabe 1: Prüfen der Differenz der Ziffern einer Zahl

Schreiben Sie ein C++-Programm, das überprüft, ob die Ziffern einer natürlichen Zahl  $n$  von links nach rechts in aufsteigender Reihenfolge sortiert sind. Das heißt, das Programm prüft, ob die Differenz zweier benachbarter Ziffern 0 oder 1 ist. Zu diesem Zweck ist die zu prüfende Zahl  $n$  einzulesen. Führen Sie die Differenzprüfung für die Ziffern der Zahl  $n$  nur durch, falls  $n \geq 100$ , andernfalls wiederholen Sie die Eingabe. Nach der Prüfung soll ausgegeben werden, ob die Differenzprüfung bestanden wurde oder nicht.

**Beispiele:** Die Ausgaben sind schwarz und die Eingaben grau dargestellt.

Geben Sie eine Zahl ein: 1234  
Differenzprüfung bestanden.

Geben Sie eine Zahl ein: 12435  
Differenzprüfung nicht bestanden.

Geben Sie eine Zahl ein: 348  
Differenzprüfung nicht bestanden.

Geben Sie eine Zahl ein: 56667  
Differenzprüfung bestanden.

(9 Punkte)

### Aufgabe 2: Ausgabe einer Pyramide als Pseudografik

Schreiben Sie ein C++-Programm, das eine Pyramide als Pseudografik mit Hilfe von verschachtelten Schleifen ausgibt. Das Programm soll die Anzahl der Zeilen für die Höhe der Pyramide erfragen und einlesen. Für die Ausgabe sind nur folgende Anweisungen erlaubt:

```
cout << " ";  
cout << "*" ;  
cout << endl;
```

**Tipp:** Geben Sie pro Zeile zuerst eine entsprechende Anzahl Leerzeichen und danach die entsprechende Anzahl Sterne aus.

**Beispiele:** Geben Sie die Höhe der Pyramide ein: 4

```
  *  
 * * *  
* * * * *  
* * * * * * *
```

Geben Sie die Höhe der Pyramide ein: 6

```
  *  
 * * *  
 * * * * *  
* * * * * * *  
* * * * * * * * *  
* * * * * * * * * *
```

(9 Punkte)