**PRO1UEGx** Übungen zu Einführung in die Programmierung, Algorithmen und Datenstrukturen **WS 25/26, Übung 8**

**Abgabetermin: Mi, 21.01.2026**

☐ **Gr. 1,** Dr. D. Auer **Name** _____ **Aufwand in h** _____
☐ **Gr. 2,** Dr. G. Kronberger
☐ **Gr. 3,** Dr. S. Wagner **Punkte** _____ **Kurzzeichen Tutor*in / Übungsleiter*in** ____ / ____

## 1. Laufzeitkomplexität (8 Punkte)

Zur Berechnung der Summe der Quadrate aller ganzen Zahlen von $a$ bis $b$ (mit $a \leq b$) werden folgende zwei Pascal-Funktionen vorgeschlagen:

```
FUNCTION SumSquares1(a, b: LONGINT): LONGINT;
  VAR
    sum, i: LONGINT;
BEGIN
  sum := 0;
  FOR i := a TO b DO BEGIN
    sum := sum + i * i;
  END; (* FOR *)
  SumSquares1 := sum
END; (* SumSquares1 *)


FUNCTION SumSquares2(a, b: LONGINT): LONGINT;
BEGIN
  IF a < b THEN
    SumSquares2 := a * a + SumSquares2(a + 1, b)
  ELSE
    SumSquares2 := a * a
END; (* SumSquares2 *)
```

a) Führen Sie eine Feinanalyse der zwei Funktionen (in Abhängigkeit von *einer* Problemgröße $n$) durch. Verwenden Sie dazu folgende relativen Ausführungszeiten:

| Operation | Laufzeit |
|---|---|
| Wertzuweisung | 1.0 |
| Vergleich | 1.0 |
| Einfache Indizierung | 0,5 |
| Addition, Subtraktion | 0.5 |
| Multiplikation | 3 |
| Division, Restbildung | 7.0 |
| Prozeduraufruf | 16 + 2 * Anz. d. Parameter |

b) Bestimmen Sie die asymptotische Laufzeitkomplexität der zwei Algorithmen und begründen Sie Ihr Ergebnis.

## 2. Zweidimensionale Felder sortieren (8 Punkte)

Gegeben ist eine Matrix mit `rows` Zeilen und `columns` Spalten auf Basis folgender Deklarationen.

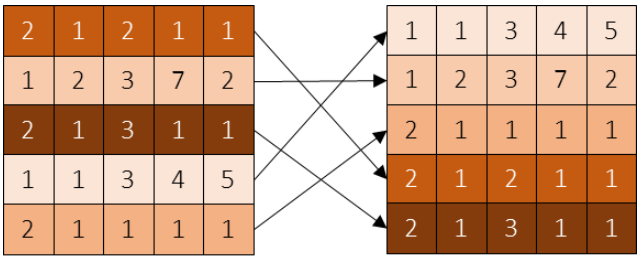
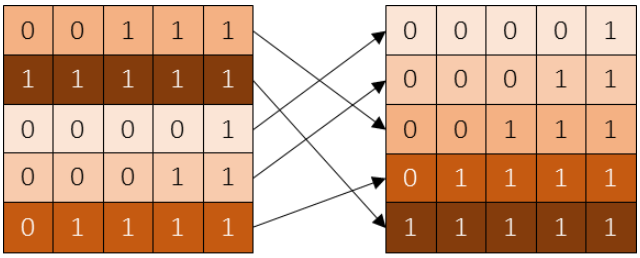
```
CONST
  rows = …;
  columns = …;
TYPE
  Row = ARRAY [1..columns] OF INTEGER;
  Matrix = ARRAY [1..rows] OF Row;
```

Gesucht ist eine Prozedur

```
PROCEDURE SortLinesByColumns(VAR m: Matrix);
```

die die Zeilen der Matrix `m` anhand der Werte in jeder Zeile (von links nach rechts) mit einem einfachen stabilen Sortierverfahren (z.B. Einfügesortieren) aufsteigend sortiert.

Beispiele:

| Beispiel | Unsortierte 5x5-Matrix | Sortierte 5x5-Matrix |
|---|---|---|
| 1 |  | |
| 2 |  | |

## 3. Auswahlsortieren auf einfach-verkettete Listen (8 Punkte)

Gegeben ist eine einfach-verkettete Liste `list` auf Basis folgender Deklarationen.

```
TYPE
  ListNodePtr = ^ListNode;
  ListNode = RECORD
    next: ListNodePtr;
    data: INTEGER;
  END; (* ListNode *)
  ListPtr = ListNodePtr;
```

Gesucht ist eine Prozedur

```
PROCEDURE SelectionSort(VAR list: ListPtr);
```

die die Liste `list` mittels Auswahlsortieren aufsteigend sortiert.

### Hinweise:

1. Geben Sie für alle Ihre Lösungen immer eine „Lösungsidee“ an.
2. Dokumentieren und kommentieren Sie Ihre Pascal-Programme.
3. Geben Sie immer auch Testfälle ab, an denen man erkennen kann, dass Ihr Pascal-Programm funktioniert, und dass es auch in Fehlersituationen entsprechend reagiert.