

Aufgabe 1

- (a) Definieren Sie die Begriffe „*partielle Korrektheit*“ und „*totale Korrektheit*“ und grenzen Sie sie voneinander ab.

partielle Korrektheit Ein Programmcode wird bezüglich einer Vorbedingung P und einer Nachbedingung Q partiell korrekt genannt, wenn bei einer Eingabe, die die Vorbedingung P erfüllt, jedes Ergebnis die Nachbedingung Q erfüllt. Dabei ist es noch möglich, dass das Programm nicht für jede Eingabe ein Ergebnis liefert, also nicht für jede Eingabe terminiert.

totale Korrektheit Ein Code wird total korrekt genannt, wenn er partiell korrekt ist und zusätzlich für jede Eingabe, die die Vorbedingung P erfüllt, terminiert. Aus der Definition folgt sofort, dass total korrekte Programme auch immer partiell korrekt sind.

- (b) Geben Sie die Verifikationsregel für die abweisende Schleife `while(B)` an.
- (c) Erläutern Sie kurz und prägnant die Schritte zur Verifikation einer abweisenden Schleife mit Vorbedingung P und Nachbedingung Q .
- (d) Wie kann man die Terminierung einer Schleife beweisen?
- (e) Geben Sie für das folgende Suchprogramm die nummerierten Zusicherungen an. Lassen Sie dabei jeweils die invariante Vorbedingung P des Suchprogramms weg. Schreiben Sie nicht auf dem Aufgabenblatt!

$$P \equiv n > 0 \wedge a_0 \dots a_{n-1} \in \mathbb{Z}^n \wedge \in \mathbb{Z}$$

```
1  i = -1;
2  // (1)
3  j = 0;
4  // (2)
5  while (i == -1 && j < n) // (3)
6  { // (4)
7      if (a[j] == m) {
8          // (5)
9          i = j;
10         // (6)
11     }
12     else {
13         // (7)
14         j = j + 1;
15         // (8)
16     }
17     // (9)
18 }
```

$$Q \equiv P \wedge (i = -1 \wedge \forall 0 \leq k < n: a_k \neq m) \vee (i \geq 0 \wedge a_i = m)$$