

**Aufgabe 1)**

```
procedure f(n, i) : integer
begin
  if (n=i)
    return n
  else
    return i * f(n, i+1);
end;
```

- a) Geben Sie die schwächste Vorbedingung an, bei dem dieses Programm terminiert.
- b) Verschärfen Sie die Vorbedingung aus a) so, dass das Programm  $n!$  berechnet.  
Beweisen Sie Ihre Aussage mit vollständiger Induktion.  
Tipp: Definieren Sie eine Invariantenbedingung, die für jedes  $i$  und  $n$  gilt und führen Sie die Induktion über die Variable  $k = n - i$
- c) Was wird berechnet wenn nur die Vorbedingung aus (a) gilt?

**Aufgabe 2)**

Untersuchen Sie die rekursive Funktion zur linearen Suche:

```
function searchLinRec (data, pos, k) : integer
begin
  if pos > length(data) then
    return 0;
  if data[pos] = k then
    return pos;
  return searchLinrec (data, pos+1, k);
end
```

- a) Formulieren Sie, was die Funktion mit diesen Parametern im Allgemeinen berechnet (also nicht nur für  $pos = 1$ ).
- b) Beweisen Sie die Behauptung von a) durch vollständige Induktion über die Anzahl der noch auszuführenden Rekursionsschritte!
- c) Begründen Sie, warum die Funktion immer stoppt!

### Aufgabe 3)

Sei  $\vec{x}$  ein Vektor aus  $n$  Zahlen ( $n \in \mathbb{N} \setminus \{0\}$ ) und  $f: (\mathbb{N})^n \rightarrow \mathbb{N}$  eine Funktion mit:

$$f(\vec{x}) = (\text{erste Komponente von } \vec{x} \text{ als Zahl}) + f(\vec{x} \text{ ohne erste Komponente}),$$

*wenn  $\vec{x}$  mehrelementig*

$$f(\vec{x}) = \text{erste Komponente von } \vec{x} \text{ als Zahl, wenn } \vec{x} \text{ einelementig}$$

Was berechnet  $f$ ? Zu welchem Rekursionstyp gehört diese Formulierung von  $f$ ?

### Aufgabe 4)

Wandeln Sie folgende rekursive Funktion in eine Funktion um, die mit einer Schleife arbeitet:

```
function f(x : R+) : R;  
begin  
  if x <= 1 then  
    return x2  
  else  
    return f(x/10);  
end;
```