## Aufgabe 1)

Betrachten Sie folgendes Programm:

```
{ n,f,k ∈ N }          φ

f := 1;
k := n;

while(k > 0) do
  begin
     k := k - 1;
     f := k • f;
  end

{ Nachbedingung }      ψ
```

a) Formulieren Sie die Invariantenbedingungen, die für jeden Schleifendurchlauf gültig sind. Beweisen Sie das durch vollständige Induktion.

b) Zeigen Sie, dass dann die Schleife irgendwann terminiert (wann genau?). Formulieren und beweisen Sie direkt unter Verwendung von a), was diese dann berechnet hat.

c) Verändern Sie die Abbruchbedingung der Schleife so, dass das Programm eine Fakultät berechnet. Welche Fakultät berechnet das Programm genau? Beweisen Sie auch das unter Verwendung von a)

## Aufgabe 2)

Betrachten Sie folgendes Programm:

```
Gegeben seien n Zahlen a[1] ... a[n] ∈ ℚ.

m := 0;
k := 0;
while (k < n) do
  begin
    m := k * m;
    k := k + 1;
    m := (m + a[k])/k ;
  end
```

a) Geben Sie eine Nachbedingung für m an! Brauchen Sie Vorbedingungen dafür?

b) Geben Sie die Invariantenbedingungen $m_i$ und $k_i$ an, die nach jedem Schleifendurchlauf erfüllt sind!

c) Geben Sie die Variantenzahl aus $\mathbb{Z}$ an, die sich in jedem Schleifendurchlauf verringert und durch welche die Schleife abbricht (nach wie vielen Schleifendurchläufen genau?), wenn sie kleiner gleich Null ist.

d) Beweisen Sie die Gültigkeit der Invariantenbedingung und den in c) festgestellten Wert der Variantenzahl mit vollständiger Induktion und folgern Sie daraus die Nachbedingung.

## Aufgabe 3)

Gegeben sei folgendes Programm:

```
{n: integer}

k := 0; s := 0;

while k ≤ n do
begin
  s := k - s;
  k := k + 1;
end {while}
```

a) Was berechnet dieses Programm? Geben Sie die genaue Abhängigkeit von n an!
b) Beweisen Sie a)!

Tipp: Unterscheiden Sie für den i-ten Schleifendurchlauf, ob i eine gerade oder eine ungerade Zahl ist.

## Aufgabe 4)

```
procedure f(n): integer

begin
   if (n = 0)
      return n
   else
      return n * f(n-1);
end;
```

**a)** Geben Sie eine Vorbedingung an, bei der dieses Programm[1] terminiert! Was berechnet das Programm unter dieser Vorbedingung?

**b)** Beweisen Sie Ihre Antwort auf die Frage in a) mit vollständiger Induktion nach n.

## Aufgabe 5) *** (für Profis)

```
{ b ∈ Z, e ∈ N }                                              φ

   result := 1;
   base := b;
   exp := e;

   while exp > 0
     begin
       if (exp MOD 2 = 1) then
         begin
           result := result * base;
           exp := exp - 1;
         end
       else
         begin
           base := base * base;
           exp := exp DIV 2;
         end;
     end; {while}
```

{ result = $b^e$ } ψ

a) Beweisen Sie mit vollständiger Induktion über k: $b^e = base_k^{\exp_k} \cdot result_k$ (Hierbei sind $base_k$, $exp_k$ und $result_k$ die Werte der Variablen nach dem k-ten Schleifendurchlauf)

b) Folgern Sie daraus, dass das Programm mit der gewünschten Nachbedingung stoppt.

[1] Die Syntax und Funktionsweise entspricht der aus dem Foliensatz GTI3Prozeduren, welche nicht mit Pascal identisch ist.