

Aufgabe 1 (6 Punkte)

Gegeben sind die folgenden zwei Funktionen `p` und `WasPassiert`.

```
function p(a:integer):boolean;

  var
    b:boolean;
    i:integer;

begin
  b := false;
  i := 2;
  while ((i < a) and (not b)) do
  begin
    b := (a mod i = 0);
    i := i + 1;
  end;
  p := (not b) and not (a < 2)
end;

function WasPassiert(a:integer):boolean;
begin
  WasPassiert := p(a) and p(a+2)
end;
```

Geben Sie eine Problemspezifikation an, die durch die Funktion `WasPassiert` gelöst wird. Wählen Sie dabei den Wertebereich der Eingabe so allgemein wie möglich.

Eingabe: x eine ganze Zahl

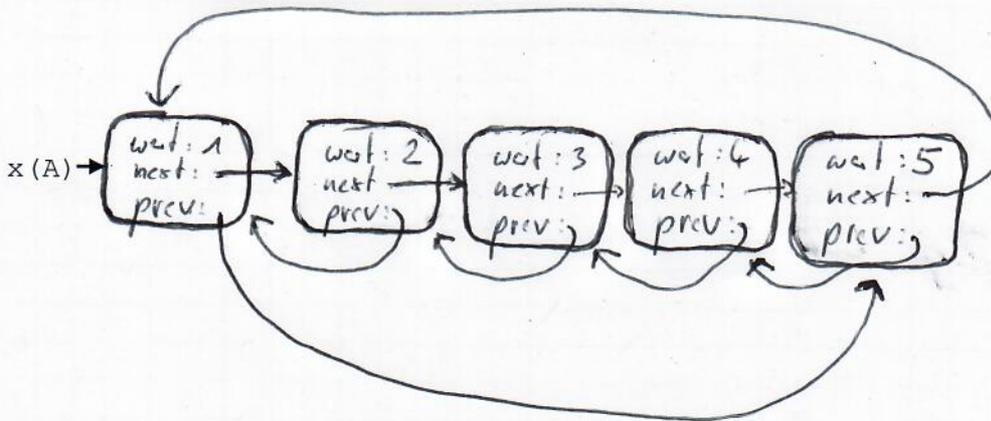
Ausgabe: b ein Wahrheitswert

Nachbedingung: b ist true \Leftrightarrow x und x+2 sind Primzahlen

Matrikelnr.: _____



Aufgabe: Welches Ergebnis liefert der Aufruf $x(A)$?



Matrikelnr.: _____

Aufgabe a: Hier finden Sie die Funktionen `c` und `x`. Die Funktion `x` ist jedoch noch unvollständig. Ergänzen Sie sie passend, an den grau eingefärbten Stellen, jeweils um eine Zeile.

```
function x(inText1:string; inText2:string; A:tRefBinBaum):integer;
```

```
  var
  i: integer;
```

```
begin
```

```
  i := 0;
```

```
  if (A=nil) then
```

```
    x := i
```

```
  else
```

```
  begin
```

```
    if (A^.text = inText1) then
```

```
      begin
```

```
        i := 1
```

```
      end;
```

```
      if (A^.text = inText2) then
```

```
        begin
```

```
          i := i - 1
```

```
        end;
```

```
        x := x(inText1, inText2, A^.li) + x(inText1, inText2, A^.rc) + i
```

```
      end
```

```
    end;
```

```
function c(inText1:string; inText2:string; A:tRefBinBaum):boolean;
```

```
begin
```

```
  c := (x(inText1, inText2, A) = 0);
```

```
end;
```

Aufgabe b: Ersetzen Sie in der Funktion `x` die Zeile `i := i - 1` durch die Zeile `i := - 1`. Geben Sie ein Testdatum an, für den die Funktion `c` Ihre Aufgabe jetzt nicht mehr erfüllt.

Sei A der Baum aus der Aufgabenstellung:
 $(aal, aal, A, true)$ wird falsch ausgewertet

Aufgabe 4 (6 Punkte)

Eine Funktion, die für eine natürliche Zahl größer als Eins die Primfaktorzerlegung bestimmt, soll einem funktionsorientierten Test unterzogen werden. Die Primfaktorzerlegung stellt die Zahl als Produkt von Primzahlen dar. Jede natürliche Zahl hat eine eindeutige Primfaktorzerlegung. Sortieren wir die Primfaktoren aufsteigend, können wir die Zerlegung eindeutig als Zeichenkette darstellen.

Beispiele: Die Primfaktorzerlegung der Zahl 18 ist $2 \cdot 3 \cdot 3$. Die Primfaktorzerlegung der Zahl 10 ist $2 \cdot 5$

Bekannt sind die folgende Typdefinition und der Funktionskopf:

type

```
tZahl = 2..maxint;
```

function faktorZerlegung(inZahl:tZahl):string;

```
{berechnet die Primfaktorzerlegung der Zahl inZahl und gibt diese  
als Zeichenkette aus.}
```

Geben Sie eine für einen Black-Box-Test sinnvolle Zerlegung der Menge der zulässigen Eingabedaten in genau drei sinnvolle Äquivalenzklassen an. Geben Sie hierbei zu jeder Äquivalenzklasse jeweils ein Testdatum an.

Klasse 1 Alle Primzahlen

Klasse 2 Alle Produkte aus mindestens zwei Primzahlen
bei denen kein Faktor doppelt vorkommt

Klasse 3 Alle Produkte aus mindestens zwei Primzahlen
bei denen ein Faktor doppelt vorkommt

Test 1 (2, 2)

Test 2 (6, 2·3)

Test 3 (18, 2·3·3)