

Übung 1: Tupel und Listen

1. Welchen Index hat das n-te Element eines Tupels oder einer Liste? Begründen Sie Ihre Aussage.

Der Index eines Tupels, einer Liste oder auch eines Arrays beginnt bei den meisten Programmiersprachen mit 0. Unter dieser Voraussetzung hat das n-te Element den Index **n-1**. Der Index einer n-elementigen sequenziellen Struktur dieser Art reicht von 0 bis n-1.

1.Element = Liste [0]
 2.Element = Liste [1]
 :
 n.Element = Liste [n-1]

Übung 2: Tupel und Listen

1. Primzahlen sind Zahlen aus dem Zahlenbereich der natürlichen Zahlen größer oder gleich 2, die außer sich selbst und der Zahl 1 keine weiteren Teiler besitzen. Alle natürlichen Zahlen lassen sich in Faktoren von Primzahlen zerlegen. Erstellen Sie einen Algorithmus, der die Primzahlen von 2 bis n mithilfe des Siebs des Eratosthenes ermittelt. Bei dieser Methode werden alle Zahlen gestrichen, die sich aus anderen Zahlen zusammensetzen.

- ✓ Schreiben Sie die Zahlen von 1 bis zur gewünschten Zahl der Größe nach geordnet in eine Liste.
- ✓ Sie starten mit der kleinsten natürlichen Primzahl, der Zahl 2.
- ✓ Setzen Sie alle Listeninhalte, die ein Vielfaches der Zahl 2 sind, auf 0.
- ✓ Ist die gesamte Liste durchlaufen, wird von der Zahl 2 aus die nächstgrößere Zahl gesucht (3).
- ✓ Alle Listeninhalte, die ein Vielfaches dieser Zahl sind, werden anschließend auf 0 gesetzt.
- ✓ Führen Sie diese Schritte so lange durch, bis das Quadrat der aktuellen Zahl größer als die letzte Zahl der Liste ist.
- ✓ In allen Listenelementen größer 1, deren Inhalt nicht auf 0 gesetzt wurde, befinden sich die gesuchten Primzahlen.

Siebzahl 2: alle Vielfache von 2 entfallen

alt	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...
neu	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...

Siebzahl 3: zusätzlich entfallen alle Vielfache von 3

alt	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...
neu	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...

usw.

Pseudocode:

```

begin sieb_des_eratosthenes
  eingabe(n);
  integer feld[n];

  for i := 0 to n-1                                // Feld mit Werten von 1 bis n belegen
  do
    feld[i] := i + 2;
  end do

  siebzahl := 2;                                     // erste Siebzahl ist 2
  while (siebzahl * siebzahl) <= n                   // Abbruchbedingung:  $s^2 \leq n$ 
  do
    p := 2;
    while (siebzahl * p) <= n                         // alle Vielfachen der Siebzahl werden
    do                                                 // gestrichen
      feld[siebzahl * p - 1] := 0;
      p := p + 1;
    end do

    do                                               // nächste Siebzahl suchen
      siebzahl := siebzahl + 1;
      while (feld[siebzahl - 1] = 0)
      end do
    end do

    for i := 1 to n-1
    do
      if feld[i] > 0 then
        ausgabe(feld[i]);                            // Ausgabe der Siebzahlen, das sind
alle                                                     // Zahlen, die größer 0 sind
      end if
    end do

end sieb_des_eratosthenes

```