

Der bestehende Iterator Ihres **ADS\_set** ist um einen zusätzlichen „Modus“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle Elemente in einer beliebigen Reihenfolge, wobei die Reihenfolge immer dieselbe sein muss, solange das **ADS\_set** nicht geändert wird (Modus „normal“). Im neuen Modus „speziell“ liefert der Iterator für ein nicht-leeres **ADS\_set** immer genau zwei Elemente, nämlich zuerst das letzte und dann das erste, bezogen auf die Iteratorreihenfolge im Modus „normal“.

**Details:** Erweitern Sie Ihre Implementierung **ADS\_set** um die Methode

```
const_iterator x() const;
```

Diese soll einen Iterator im Modus „speziell“ liefern. Wenn kein Element im **ADS\_set** vorhanden ist, dann gilt **x() == end()**. Andernfalls liefert **x()** einen Iterator, der auf das letzte Element zeigt. Der von **x()** gelieferte Iterator soll nach dem ersten Inkrementieren auf das erste Element (bezogen auf die normale Iteratorreihenfolge der Elemente) zeigen, nach dem zweiten Inkrementieren wird **end()** geliefert. Achtung: wenn das **ADS\_set** genau ein Element enthält, dann ist dieses Element das erste Element und zugleich das letzte Element und wird daher vom Iterator zwei Mal geliefert.

Die Speicherkomplexität der Operatorfunktionen muss unverändert bleiben (so sind zB zusätzliche Felder mit nicht konstanter Größe unzulässig). Für die Zeitkomplexität gilt grundsätzlich dasselbe, das lineare Durchlaufen der Struktur zum Positionieren des Iterators ist aber natürlich zulässig.

Beispiele:

Angenommen der von <b>begin()</b> retournierte Iterator liefert alle gespeicherten Elemente in der Reihenfolge	dann liefert der von <b>x()</b> retournierte Iterator die folgenden Elemente in der Reihenfolge
(4,7,1,5,3,6,0,8,10,2,9)	(9, 4)
(7,8,9,4,1,5,10)	(10, 7)
(7,8,9,4,1,5)	(5, 7)
(7,8,9)	(9, 7)
(7,9)	(9, 7)
(9)	(9, 9)
()	()

**Anleitung:** Schreiben Sie **keine** neue Iteratorklasse!

Erweitern Sie die bestehende Iterator-Klasse wie folgt (dies ist nur einer der möglichen Lösungsansätze, abweichende korrekte Lösungen sind natürlich zulässig):

- Es muss ein Iterator im Modus „speziell“ erzeugt werden können. Dazu ist ein neuer Konstruktor zu schreiben und/oder bestehende zu erweitern, um die Instanzvariablen entsprechend zu initialisieren. Bei der Erzeugung wird der Iterator auf das letzte Element gesetzt, indem zB alle anderen Elemente übersprungen werden.
- Passen Sie die Inkrement-Operationen so an, dass der Iterator im Modus „speziell“ beim ersten Inkrementieren auf das erste Element und beim zweiten Inkrementieren auf **end()** gesetzt wird. Dazu kann es hilfreich sein, wenn schon beim Erzeugen des Iterators entsprechende Informationen (zB Positionen) gespeichert werden.

Die Methode **ADS\_set::x()** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen.

Die Methode **ADS\_set::begin()** liefert wie bisher einen Iterator im Modus „normal“.