

Der bestehende Iterator Ihres **ADS\_set** ist um einen zusätzlichen Modus „speziell“ 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 Modus speziell liefert der Iterator die Elemente in derselben relativen Reihenfolge gemäß folgender Bildungsregel (die Nummerierung der Elemente entspricht der Reihenfolge im Modus „normal“): zuerst wird das 1. Element geliefert, dann das 1. und 2. Element, dann das 1., 2. und 3. Element usw. bis am Ende alle Elemente geliefert werden. Für ein **ADS\_set** der Größe  $n$  liefert der Iterator im Modus speziell somit insgesamt  $n*(n+1)/2$  Elemente, nach dem  $i*(i+1)/2$ -ten Element wird wieder von vorne begonnen, wobei  $i = 1..n-1$

In beiden Modi erreicht der Iterator nach dem letzten Element **end()**.

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

**const\_iterator y() const;**

Diese soll einen Iterator im Modus „speziell“ erzeugen. Wenn kein Element im **ADS\_set** vorhanden ist, dann gilt **y() == end()**. Andernfalls liefert **y()** einen Iterator, der auf das 1. Element zeigt und beim Inkrementieren gemäß obiger Bildungsregel entweder auf das nächste Element oder wieder auf das 1. Element gesetzt wird.

Die Zeitkomplexität und Speicherkomplexität der Operatorfunktionen müssen unverändert bleiben. So sind z. B. zusätzliche Felder mit nicht konstanter Größe unzulässig. Beispiele:

Angenommen der von <b>begin()</b> retournierte Iterator liefert alle gespeicherten Elemente in der Reihenfolge	Dann liefert der von <b>y()</b> retournierte Iterator die folgenden Elemente in der Reihenfolge
(1,2,3,4)	(1,1,2,1,2,3,1,2,3,4)
(9,7,8)	(9,9,7,9,7,8)
(7,9)	(7,7,9)
(7)	(7)
()	()

**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. Eventuell benötigen Sie zusätzliche Instanzvariablen, um den Anfangszustand zu speichern.
- Passen Sie die Inkrement-Operationen (nur!) für den Modus „speziell“ an. Dieser soll

Die Methode **ADS\_set::y()** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen. Die Methode **ADS\_set::begin()** liefert wie bisher einen Iterator im Modus „normal“.