

Der bestehende Iterator Ihres **ADS_set** ist um einen zusätzlichen Modus „speziell“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle n Elemente des **ADS_set** 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 Reihenfolge, allerdings werden die ersten $\left\lfloor \frac{n}{4} \right\rfloor$ Elemente und die letzten $\left\lfloor \frac{n}{4} \right\rfloor$ Elemente (das erste Viertel und das letzte Viertel der Elemente, abgerundet) übersprungen. In beiden Modi erreicht der Iterator nach dem letzten gelieferten Element **end()**.

Details: Erweitern Sie Ihre Implementierung **ADS_set** um die Methode

const_iterator x() const;

x() erzeugt und retourniert einen Iterator im Modus „speziell“, der genau $n - 2 \left\lfloor \frac{n}{4} \right\rfloor$ Werte liefert.

Die Zeitkomplexität und Speicherkomplexität aller Funktionen müssen unverändert bleiben. So sind z. B. zusätzliche Felder mit nicht konstanter Größe unzulässig. Das Überspringen von Werten im Iterator, zB mit einer Schleife, ist erlaubt.

Beispiele:

Angenommen der von begin() retournierte Iterator liefert alle n gespeicherten Elemente in der Reihenfolge	Dann liefert der von x() retournierte Iterator die folgenden Elemente in der folgenden Reihenfolge
(1,2,3,4,5,6,7,8)	(3,4,5,6)
(1,2,3,4,5,6,7)	(2,3,4,5,6)
(4,3,6,1)	(3,6)
(9,7,8)	(9,7,8)
(7,9)	(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 sind zu erweitern, um die Instanzvariablen entsprechend zu initialisieren. Eventuell benötigen Sie zusätzliche Konstruktorparameter und/oder Instanzvariablen, um zu speichern, wie viele Elemente zu liefern sind und wie viele Elemente zu überspringen. Beim Erzeugen des Iterators sind unter Umständen bereits Elemente zu überspringen.
- Passen Sie die Inkrement-Operationen (nur!) für den Modus „speziell“ an: nach dem letzten zu liefernden Element wird der Iterator auf **end()** gesetzt.

Die Methode **ADS_set::x()** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen. Es kann sinnvoll sein, schon in **x()** zu ermitteln, wie viele Elemente zu liefern und wie viele zu überspringen sind, um diese Werte an den Konstruktor des Iterators zu übergeben.

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