Lesen Sie diese Seite bis zum Ende durch!

Drehen Sie das Blatt erst um, wenn die Prüfungsaufsicht den Start der Prüfung verkündet!

Vorgangsweise bei der Prüfung

- Studierendenausweis auf dem Platz hinterlegen
- Elektronische Geräte aller Art ausschalten und unerreichbar verstauen.
- Moped Kiosk starten
- Einloggen
- Lesen und Beachten der Regeln, die auf der Webseite angeführt sind.
- Änderungen in Ihrem Programm durchführen (Edit Button)
- Hochladen und Testen (Abschicken Button)
- Sichern ohne Test ist ebenfalls möglich (Speichern Button rechts oben)
- Um mit Ctrl-F im Programmtext zu suchen, müssen Sie zuerst in das Editorfenster klicken, da sonst eventuell die Browsersuche aktiviert wird.
- Hilfreiche Tastenkombination: Alt-F4 (Fenster schließen) Achtung: nicht gespeicherte Änderungen gehen verloren
- Sobald Sie SUCCESS erreichen, melden Sie sich bitte **unbedingt** bei der Prüfungsaufsicht, bevor Sie Ihren Arbeitsplatz verlassen. Andernfalls ist Ihr Ergebnis ungültig.
- Wenn kein SUCCESS erreicht wird, kann die Klausur zum nächsten Termin wiederholt werden.
- Programme, die SUCCESS erreichen, werden noch auf Korrektheit (z.B. Ordnungsverhalten) geprüft. Ihr Projekt wird ebenfalls noch einem Korrektheits- (richtige Datenstruktur, Einhaltung der Spezifikation) und einem Plagiatscheck unterzogen. Falls all diese Prüfungen auch erfolgreich durchlaufen werden, werden die Punkte für das Projekt vergeben.

Jede Verwendung zusätzlicher Hilfsmittel bzw. Kommunikation mit Nachbar*innen wird als Versuch gewertet, die Leistung zu erschleichen.

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" sollen die Werte in derselben Reihenfolge geliefert werden, aber es wird nach jedem gelieferten Wert eine bestimmte Anzahl von Werten übersprungen. In beiden Modi erreicht der Iterator **end()**, sobald keine Elemente mehr vorhanden sind.

Details: Erweitern Sie Ihre Implementierung ADS_set um die Methode

```
const_iterator v(size_t n) const;
```

Diese soll einen Iterator im Modus "speziell" liefern. Der Parameter \mathbf{n} gibt an, wie viele Werte nach jedem Wert zu überspringen sind. Wenn kein Element im $\mathtt{ADS_set}$ vorhanden ist, dann gilt $\mathbf{v}(\) == \mathtt{end}(\)$. Der von $\mathbf{v}(\)$ gelieferte Iterator soll beim Inkrementieren in weiterer Folge jeweils \mathbf{n} Werte überspringen und auf den darauffolgenden Wert zeigen, falls ein solcher vorhanden ist, andernfalls wird $\mathtt{end}(\)$ geliefert.

Die Zeitkomplexität und Speicherkomplexität der Operatorfunktionen müssen unverändert bleiben. So sind zB zusätzliche Felder mit nicht konstanter Größe unzulässig.

Beispiel: Angenommen der normale Iterator Ihres **ADS_set** liefert alle gespeicherten Elemente in der Reihenfolge (4,7,1,3,6,8,10,2), dann gilt

- v(0) liefert einen Iterator, der auf 4 zeigt.
 Wenn der Iterator bis end() inkrementiert wird, liefert er die Folge (4,7,1,3,6,8,10,2)
- v(1) liefert einen Iterator, der auf 4 zeigt.
 Wenn der Iterator bis end() inkrementiert wird, liefert er die Folge (4,1,6,10)
- v(2) liefert einen Iterator, der auf 4 zeigt.
 Wenn der Iterator bis end() inkrementiert wird, liefert er die Folge (4,3,10)
- v(10) liefert einen Iterator, der auf 4 zeigt.
 Wenn der Iterator bis end() inkrementiert wird, liefert er die Folge (4)

Anleitung: Schreiben Sie **keine** neue Iteratorklasse! Erweitern Sie die bestehende Iterator-Klasse wie folgt:

- Es ist zu speichern, ob der Iterator im Modus "normal" oder "speziell" ist. Für den Modus "speziell" muss gespeichert werden, wie viele Werte bei jeder Inkrementierung zu überspringen sind.
- 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.

Passen Sie die Inkrement-Operationen so an, dass entsprechend viele Werte übersprungen werden, wenn der Iterator im Modus "speziell" ist.

Die Methode ADS_set::v() erzeugt einen Iterator im Modus "speziell" und retourniert diesen. Die Methode ADS_set::begin() liefert wie bisher einen Iterator im Modus "normal".