

Aufgabe

Die bisher implementierte Version des Vektor-Moduls stellt dem Anwendungsprogramm einen statischen Vektor zur Verfügung, d.h. die Größe des Vektors muss bereits zur Übersetzungszeit festgelegt werden.

Da sich die Anzahl der benötigten Vektorelemente in vielen Fällen erst zur Laufzeit herausstellt, soll eine weitere Version des Vektor-Moduls implementiert werden, die einen dynamisch erweiterbaren Vektor bereitstellt.

Der Vektor soll zur Laufzeit mit einer fest vorgegebenen Anfangslänge (z.B. 16) erzeugt werden. Immer dann, wenn ein Element in den Vektor eingefügt werden soll und für dieses Element im Vektor kein Speicherplatz mehr zur Verfügung steht, also `size() == capacity()` gilt, soll die Größe des Vektors automatisch verdoppelt werden. Der Wert von `size()` und die in dem Vektor gespeicherten Elemente dürfen durch dieses Vergrößern nicht verändert werden. Jedoch muss `capacity()` anschließend die neue Länge des Vektors liefern.

Darüber hinaus soll dem Anwendungsprogramm eine weitere Schnittstellenfunktion zur Verfügung gestellt werden:

```
void reserve( unsigned n );
```

Diese Funktion soll dem Anwendungsprogramm die Möglichkeit geben, den Vektor auf die Länge `n` einzustellen. (Falls das Anwendungsprogramm z.B. die maximal benötigte Größe des Vektors kennt, ist es in der Regel effizienter einmal `reserve(maxlen)` aufzurufen, als sich auf den Vergrößerungsautomatismus zu verlassen.)

Da mit Hilfe von `capacity()` überprüft werden kann, ob der Aufruf von `reserve` erfolgreich war, braucht die Funktion keinen Resultatwert zurückgeben.

Falls `n` kleiner als `size()` ist, soll der Vektor anschließend auch nur noch die ersten `n` Elemente enthalten.

`reserve(0)` soll dafür sorgen, dass der gesamte Speicherplatz, der für den Vektor angelegt ist, freigegeben wird, d.h. anschließend soll `capacity() == 0` gelten.

Zusätzlich soll das Modul dem Anwendungsprogramm eine globale Variable `debug_vektor` zur Verfügung stellen, die zu Beginn der Programmausführung den Wert 0 (*false*) hat. Wenn diese Variable durch das Anwendungsprogramm auf einen Wert ungleich 0 (*true*) gesetzt ist, soll bei jedem (Neu-)Anlegen des Vektors eine Information auf der Standardfehlerausgabe ausgegeben werden, die die alte und neue Größe des Vektors beinhaltet.

Mit Hilfe dieser globalen Variablen wird dem Anwender also ein gewisser Einblick in die Funktionsweise des Moduls gegeben.

Zum Testen Ihres neuen Vektor-Moduls benutzen Sie bitte das Programm `Decrypt` (Lesen von verschlüsseltem Text).