

Aufgabe 1 – Merge

Erstellen Sie eine Funktion `merge(...)`, die zwei sortierte Vektoren zusammenführt und das sortierte Ergebnis in einem dritten Vektor ablegt. Die Vektorelemente weisen den Typ `int` auf und sind in aufsteigender Reihenfolge sortiert. Gehen Sie davon aus, dass der dritte Vektor groß genug ist, um alle Elemente beider Vektoren zu speichern.

Die Funktion erwartet als Argumente die beiden sortierten Vektoren, ihre Längen sowie den dritten Vektor. Sie liefert nach dem Zusammenführen die Anzahl Elemente im dritten Vektor als Return-Wert oder `-1`, falls einer der beiden übergebenen Vektoren nicht in aufsteigender Reihenfolge sortiert ist.

Der Prototyp der Funktion `merge` ist vorgegeben und sieht wie folgt aus:

```
int merge(int* v1, int v1Len, int* v2, int v2Len, int* vDest);
```

Verwenden Sie Zeiger, um die auf die einzelnen Elemente der Vektoren zuzugreifen.

Also z.B. so (`v1` sei ein Vektor vom Typ `int`, `v1len` die Länge desselben):

```
for (int* ptr = v1; (ptr - v1) < v1len; ptr++)
{
    cout << *ptr << endl;
}
```

Und nicht so:

```
for (int i = 0; i < v1len; i++)
{
    cout << v1[i] << endl;
}
```

(Im Ergebnis machen die beiden Schleifen im Übrigen das gleiche...)

Lagern Sie die Implementierung von `merge(...)` in eine Header-Datei Namens `,vectorlib'` aus.

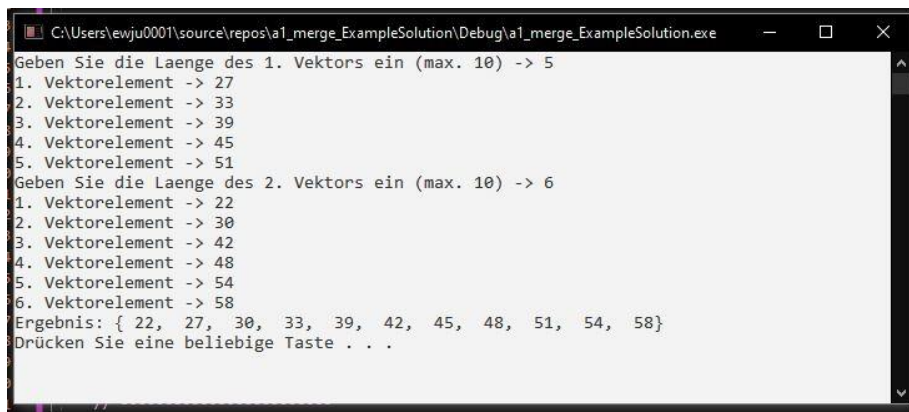
Zum Testen schreiben Sie ein Programm:

Der Benutzer gibt zunächst die beiden Vektoren (inkl. deren Längen) über die Tastatur ein. Mit diesen beiden Vektoren wird `merge(...)` aufgerufen. Im Anschluss daran wird dann der Ergebnis-Vektor auf dem Bildschirm ausgegeben, falls der Rückgabewert ≥ 0 ist (siehe oben). Andernfalls informieren Sie den Anwender darüber, dass einer der beiden Eingangs-Vektoren nicht sortiert war.

ACHTUNG: Aus der Funktion `merge(...)` heraus erfolgen keine Ausgaben auf dem Bildschirm!!!

Wichtig: Die beiden Vektoren, die Sie als Eingabe-Parameter an `merge(...)` übergeben, dürfen selbst nicht verändert werden!

Beispiel (BildschirmAusgabe):



```
C:\Users\ewju0001\source\repos\al_merge_ExampleSolution\Debug\al_merge_ExampleSolution.exe
Geben Sie die Laenge des 1. Vektors ein (max. 10) -> 5
1. Vektorelement -> 27
2. Vektorelement -> 33
3. Vektorelement -> 39
4. Vektorelement -> 45
5. Vektorelement -> 51
Geben Sie die Laenge des 2. Vektors ein (max. 10) -> 6
1. Vektorelement -> 22
2. Vektorelement -> 30
3. Vektorelement -> 42
4. Vektorelement -> 48
5. Vektorelement -> 54
6. Vektorelement -> 58
Ergebnis: { 22, 27, 30, 33, 39, 42, 45, 48, 51, 54, 58}
Drücken Sie eine beliebige Taste . . .
```

Hinweis 1:

Prüfen Sie in `merge(...)` zuerst, ob die beiden übergebenen Vektoren bereits sortiert sind. Um sich die Arbeit zu erleichtern, schreiben Sie hierfür nicht zweimal denselben Code, sondern? Richtig: Sie schreiben eine eigenen Funktion, z.B. `bool isSorted(...)`. Genauso könnte es sich anbieten, auch für weitere Aufgaben eigene Funktionen zu schreiben (z.B. Ausgabe eines Vektors auf dem Bildschirm, Anhängen eines Vektors an einen anderen, ...). Das macht Ihr Programm verständlicher und übersichtlicher, bricht die Komplexität herunter und erleichtert Ihnen die Arbeit, wenn Sie die Funktionen mehrmals benötigen oder Sie diese später in einem anderen Programm wiederverwenden möchten.

Hinweis 2:

Machen Sie sich die Tatsache zunutze, dass beide Eingabe-Vektoren bereits sortiert sind.

Beispiel:

	Erster Vektor	Zweiter Vektor	Ergebnisvektor
1.	27 33 39 45 51	22 30 42 48 54 58	22
2.	27 33 39 45 51	22 30 42 48 54 58	22 27
3.	27 33 39 45 51	22 30 42 48 54 58	22 27 30
4.	27 33 39 45 51	22 30 42 48 54 58	22 27 30 33
...			

Abgabe:

Zusammen mit Ihrem **vollständig kommentierten und getesteten** Projekt geben Sie 3 Screenshots wie oben (unter Windows 10 mit ‚Snipping Tool‘ möglich) mit folgenden Vektoren ab. Komprimieren Sie die vollständige Projektdatei zusammen mit den Screenshots in eine zip-Datei.

1. $v1 = \{ 10, 20, 30, 40, 50 \}$
 $v2 = \{ 25, 31, 33, 40, 50, 60, 70 \}$
2. $v1 = \{ 25, 31, 33, 40, 50, 60, 70 \}$
 $v2 = \{ 10, 20, 30, 40, 50 \}$
3. $v1 = \{ 25, 31, 33, 40, 50, 60, 70 \}$
 $v2 = \{ 10, 20, 30, 50, 40 \}$

Viel Spaß und viel Erfolg!