



# Übungsblatt 1: Verkettete Liste

Abgabetermin: 06.11.2025 23:59 Uhr

## Allgemeine Hinweise zu den BS Übungen

- Die Aufgaben sind alleine zu bearbeiten, allerdings ist Partnerarbeit (maximal zu zweit) möglich. Beide Partner müssen aber in ihrem Git-Repository abgeben und eine PARTNER-Datei mit dem Name des Partners im Repository anlegen.
- Jeder Teilnehmende muss dabei in der Lage sein, die gesamte Lösung zu erläutern.
- Zur Versionsverwaltung wird jedem Teilnehmenden ein Git-Repository zur Verfügung gestellt, welches auch zur Abgabe verwendet wird. Der Zugriff auf das Repository erfolgt über das IBR GitLab unter <https://gitlab.ibr.cs.tu-bs.de>.
- Die Abgabe ist spätestens zur Deadline durch ein `git push` zu tätigen. Es sollte sichergestellt werden, dass die Änderungen vorher auch durch ein `git commit` gesichert wurden.
- Eine Aufgabe kann beliebig oft abgegeben werden; es gilt die letzte *rechtzeitige* Abgabe.
- Die abgegebenen Programme werden auf Ähnlichkeit mit anderen Programmen desselben Semesters und früherer Semester überprüft. Bei starken Übereinstimmungen behalten wir uns ein Ausschluss vom Übungsbetrieb und eine Meldung an den Prüfungsausschuss vor.
- Die Lösungen der Programmieraufgaben müssen im GitLab CI/CD lauffähig sein. Zusätzlich kann das System `x1.ibr.cs.tu-bs.de` zum Testen genutzt werden. Auf diesem können Sie sich per SSH mit Ihrem IBR Account anmelden.
- Für die Bearbeitung kann auf einem beliebigen Linux-System gearbeitet werden. Allerdings empfehlen wir das frühzeitige Testen der Abgabe durch ein `git push` und die Inspektion der Test Pipeline im GitLab.
- Die Programmieraufgaben werden automatisch durch das bereitgestellte Testsystem per GitLab CI/CD ausgewertet. Es müssen mindestens 50 % der verfügbaren Testcase-Punkte erreicht werden.
- Die Testcase-Punkte können kontinuierlich über die Ausgabe des `run-tests` Jobs aus der GitLab CI/CD Pipeline eingesehen werden. Gewertet wird der letzte Pipeline-Status auf dem von uns zur Abgabe angelegten Abgabebranch (z.B. `abgabe1`), überprüfen Sie diesen gegebenenfalls.
- Sollten Sie 50 % der Punkte erreicht haben werden Sie ggf. für ein Code-Interview ausgewählt. Sollten Sie zum Code-Interview nicht erscheinen oder ihre Lösung nicht ausreichend erklären können wird der Zettel als „ungenügend“ bewertet.
- **KI Richtline:** Der Einsatz von KI-gestützten Tools ist bei der Bearbeitung der Hausaufgaben streng untersagt, da die eigenständige Erarbeitung der Lösungen inhärentes Lernziel ist. Es ist zu beachten, dass wir alle Abgaben als Prüfungsdokumente archivieren und es auch bei einer nachträglichen Prüfung zu einer rückwirkenden Aberkennung der Studienleistung kommen kann.

## Aufgabe 1: Verkettete Liste

Implementieren Sie eine einfach verkettete Liste, welche nicht-negative Ganzzahlen verwaltet. Auf die Liste soll mit den folgenden Funktionen zugegriffen werden:

- `int list_append(list_t *list, int value)`: Fügt einen Wert in die Liste ein, wenn dieser noch nicht vorhanden ist. Im Erfolgsfall gibt die Funktion den eingefügten Wert zurück, ansonsten den Wert -1.
- `int list_pop(list_t *list)`: Entnimmt den ältesten Wert aus der Liste und gibt diesen zurück. Ist kein Wert in der Liste vorhanden, wird -1 zurückgeliefert.

Im Repository finden Sie eine Vorlage für die Quelldatei (`lilo.c`). Das darin enthaltende Hauptprogramm (Funktion `main()`) fügt einige Werte in die Liste ein und entnimmt diese wieder. Die Textsequenz aus der Vorlage soll folgende Ausgabe erzeugen:

```
insert 47: 47
insert 11: 11
insert 23: 23
insert 11: -1
remove: 47
remove: 11
```

### Hinweis:

- Hilfreiche Manual-Pages: **malloc(3)**, **free(3)**
- Implementieren Sie keine Listenfunktionalität in der Funktion `main()`. Allerdings können Sie die Funktion `main()` erweitern, um Ihre Implementierung zu testen.
- Der Versuch, eine negative Zahl in die Liste einzufügen, soll unterbunden und als Fehler gewertet werden.
- Sollte bei der Ausführung einer verwendeten Funktion (z.B. **malloc(3)**) ein Fehler auftreten, sind keine Fehlermeldungen auszugeben; der Fehler muss aber gemäß der Aufgabenstellung sinnvoll behandelt werden.
- Sie können den Code mit folgendem Aufruf übersetzen:  
`gcc -std=c11 -pedantic -Wall -Wextra -Werror -o lilo lilo.c`