

Praxis der Programmierung

Hausaufgabenprojekt zur Programmierung in C

Aufgaben:

1. Implementieren Sie eine einfach verkettete Liste mit den Typen `listelement` und `list` aus der Vorlesung mit der folgenden Anpassung: Der Wert eines Listenelements ist ein ganzzahliges Array der Größe `N` (statt einem einzelnen ganzzahligen Wert). `N` wird dabei von einer symbolischen Konstante am Anfang des Programms als 5 definiert. Implementieren Sie die folgenden Funktionen:
 - (a) `void insert(int ar[], list * l)`: Ein neues Element mit dem Wert `ar` am Listenanfang angefügt.
 - (b) `void delete_all(list * l)`: Die gesamte Liste wird gelöscht.
 - (c) `void print_list(list l)`: Die gesamte Liste wird auf der Konsole ausgegeben. Dabei sind einzelne Listenelemente mit Komma getrennt und die Arrays werden in der Form `[1,2,3,4,5]` ausgegeben.

Halten Sie sich soweit wie möglich an die Definitionen aus der Vorlesung!

2. Ergänzen Sie eine Funktion `int compare(int ar[], list l, int pos)`, die ermittelt, ob das Listenelement an Position `pos` die Liste `ar` als Wert enthält. Dazu werden die Arrays elementweise verglichen. Wenn die Arrays nicht übereinstimmen, wird der erste Index, an dem sie sich unterscheiden, zurückgegeben. Stimmen die Arrays in allen Elementen überein, ist der Rückgabewert `-1`. Wenn die Liste weniger als `pos` Elemente hat oder leer ist, gibt die Funktion `-2` zurück.
3. Ergänzen Sie eine Funktion `int insertBehind(int new[], list * l, int ar[])`, die ein Element mit dem Array `new` als neues Element hinter dem ersten Listenelement, das das Array `ar` als Wert enthält, einfügt. Zum Vergleichen der Arrays wird die Funktion `compare` genutzt. Existiert das Listenelement, hinter dem das Element eingefügt werden soll, nicht, wird `-1` zurückgegeben, sonst `0`.
4. Ergänzen Sie eine Funktion `list reverse(list l)`, die eine Liste übergeben bekommt, eine neue Liste erzeugt, in der die Reihenfolge der Listenelemente umgekehrt ist, und diese neue, umgekehrte Liste zurückgibt. Dabei soll in jedem Element auch das Werte-Array umgekehrt werden.
Beispiel: `[1,2,3,4,5]`, `[6,7,8,9,10]` wird zu `[10,9,8,7,6]`, `[5,4,3,2,1]`.
5. Definieren Sie in der `main`-Funktion unter Verwendung der Funktion `insert` eine Liste vom oben genannten Typ `list` mit mindestens drei Listenelementen.
6. Testen Sie alle implementierten Funktionen mit dieser Liste und mit der leeren Liste.

Allgemeine Anforderungen an Abgaben:

- Die Lösungen müssen bis Freitag, den 20. Juni 2025, um 23:59 Uhr auf Moodle hochgeladen werden. Wir empfehlen die Aufgabe innerhalb der Pfingstwoche zu bearbeiten und abzugeben.
- Der Kommentarblock am Anfang jedes Quellcodes muss Ihren Namen und Ihre Matrikelnummer enthalten.
- Alle C-Programme müssen auf dem Linux-Server `ts-linux.cs.uni-potsdam.de` ausführbar sein. Wir empfehlen dies vor der Abgabe mindestens einmal zu testen (z.B. im PC-Pool). Programme, die nicht ausführbar sind oder deren Definitionen (insb. Aufgabe 1) nicht mit der Vorlesung oder den hier genannten Vorgaben übereinstimmen, werden sofort mit 'nicht bestanden' bewertet!
- Neben den auf diesem Blatt gelisteten Anforderungen, können Sie bspw. ergänzende Funktionen implementieren, um die Implementierung der geforderten Funktionen zu vereinfachen.
- Das Programm soll auch noch lauffähig sein, wenn die symbolische Konstante `N` mit einem anderen Wert (bspw. 10) definiert wird.
- Kommentieren Sie Ihren Code angemessen.