Häufige Fehler Klausur 2abc sengT

a.
Korrekterweise muss in einer create-Funktion nach versuchter Allokation getestet werden, ob Memory auf dem Heap auch tatsächlich erhalten wurde. Geschieht dies über mehrere Stufen, muss dies stets von neuem getestet werden. Für den Fall des Misslingens muss bisher alloziertes Memory wieder freigegeben werden.

b.
Beim Allozieren soll nie auf einen Pointer ungleich NULL getestet werden und allenfalls free ausgeführt werden. Beim Allozieren ist jedes free überflüssig und kann sogar schädlich sein, wenn nicht allozierter Platz freigegeben wird. Dieser Fehler geschieht gerne implizite beim Aufrufen einer Funktion.

c.
Haben Sie für ein Array Platz alloziert, wird das gesamte Array in einem Mal freigegeben, nicht Slot-weise.

d.
Wird zuerst Memory freigegeben, auf das die Pointer im Array zeigen und gleich anschliessend das Array selbst freigegeben, so brauchen die Slots des Array icht NULL gesetzt zu werden.

e.
Das NULL-Setzen eines Pointers in einer Funktion, der als Paramter übergeben wurde, bringt i.A. nichts. Es wird ja nur die Kopie mit NULL initialisiert. Statt dessen muss ein Pointer auf ein Pointer eines Typs verwendet werden.

f.
Statt z.B mit Node** node können Sie auch jederzeit mit Node *node[] arbeiten. Dies gilt für formale Parameter wie beim Aufruf (hier ist ein Index anzugeben).

g.
Entscheiden Sie sich, ob sie mit struct _Node oder mit Node (vorher mit entsprechendem typedef definiert) arbeiten wollen. Vermeiden Sie ein Mischen der beiden. Gar nicht funktionieren kann
struct Node.

h.
Strings müssen mit strcpy kopiert werden. Direkte Zuweisung mit = funktioniert in C nicht, bzw. nur am Ort der Deklaration, was bei dynamischen Lösungen nicht möglich ist.

Bemerkenswert

i.
realloc kann auch als malloc verwendet werden. Dazu ist der erste Parameter NULL zu setzen. Mit dieser Kenntnis kann u.U. Code eleganter und kürzer formuliert werden.