

1 Dynamische Summen-Pyramide

Ziel dieser Übung ist es, eine visuelle Darstellung einer Summen-Pyramide (vergleichbar mit einem invertiertem Pascalsches Dreieck) zu entwickeln.

Eingabe

Als Eingabe soll der User zur Laufzeit zunächst die Zahl der Summanden (maximal 10) und dann nachfolgend die einzelnen Summanden eingeben (vgl. listing 1).

```
1 Gesamtzahl Summanden:  
2 6  
3 Summanden:  
4 1  
5 1  
6 2  
7 4  
8 1  
9 9
```

Listing 1: "Beispiel-Input"

Ausgabe

Auf jeder nachfolgenden Ebene wird aus zwei benachbarten Summanden die Summe gebildet und als neuer Summand eingetragen (vgl. listing 2). Es kann auch ein vereinfachtes Layout verwendet werden, solange die Abfolge der Iterationen und Summanden erkennbar ist.

```
1      1      1      2      4      1      9  
2         2      3      6      5      10  
3            5      9      11     15  
4               14     20     36  
5                  34     56  
6                     90
```

Listing 2: "Beispiel-Output"

Verwenden Sie für die Implementierung dynamischen Speicher und speichern Sie die das Ergebnis zunächst zwischen, bevor Sie es ausgeben. Sie können es sich aussuchen, ob zunächst die komplette Summen-Pyramide berechnet und gespeichert werden soll oder ob das Ergebnis zeilenweise gespeichert und dann ausgegeben werden soll.

Verwenden Sie als Vorlage `03_valgrind_exercise_02.c`. Beachten Sie, dass Sie die Methoden-Signatur der Einlese-Methode auf Ihre Bedürfnisse anpassen müssen. Verwenden Sie Valgrind, um Ihre Implementation auf Speicherfehler zu testen.

2 Dynamischer Struct-Speicher

Implementieren Sie einen dynamischen Struct-Speicher, in dem Koordinaten-Structs abgelegt, ausgelesen und wieder entfernt werden können. Es ist Ihnen überlassen, ob Sie für die Implementierung der Datenstruktur sich an einem Array oder einer LinkedList oder etwas ganz anderem orientieren.

Folgende Funktionen soll Ihre Datenstruktur unterstützen:

- Erstellen und Freigeben Ihrer neuen Struct-Speicher-Datenstruktur zur Speicherung der Koordinaten-Structs
- Hinzufügen eines Koordinaten-Structs an beliebiger Position oder am Ende
- Ausgabe der Koordinaten x , y , z eines einzelnen Koordinaten-Structs, aller Koordinaten-Structs in einem Bereich und aller Koordinaten-Structs in der Struct-Speicher-Datenstruktur auf die Konsole
- Löschen eines einzelnen Koordinaten-Structs, aller Koordinaten-Structs in einem Bereich und aller Koordinaten-Structs
- Ausgabe der Anzahl an Koordinaten-Structs in Ihrer Struct-Speicher-Datenstruktur

Verwenden Sie als Vorlage `03_valgrind_exercise_03.c` und überprüfen Sie Ihre Applikation mit Valgrind.

Hinweis 1: Viele Methoden können durch einen angepassten Aufruf einer ähnlichen Methode implementiert werden; Sie können damit Zeit und Arbeit sparen.

Hinweis 2: Um den Compiler zu unterstützen, können Sie die void-Pointer in den Methoden-Signaturen, die auf die dynamische Struct-Speicher-Datenstruktur zeigen, durch Pointer auf Ihre eigene Datenstruktur ersetzen.

Bonus:

- Verwenden Sie `callgraph`, um Ihren Programmablauf zu visualisieren.
- Verwenden Sie `massif`, um die Verwendung des Heap-Speichers durch Ihr Programm zu visualisieren.