

Was ist eine header-Datei ? Was darf in einer header-Datei nicht drinnen stehen ?

Header-Dateien enthalten Informationen, die in mehreren C-Dateien gebraucht werden. Definitionen von Präprozessorkonstanten, Typ- und Strukturdefinitionen und Funktionsdeklarationen. Header-Dateien dürfen keine ausführbaren Programmtexte (Definition von Funktionen) enthalten. Header-Dateien werden auch mit Bibliotheken mitgeliefert.

Was ist der Unterschied zwischen der Deklaration und Definition einer Variable?

Unter der Deklaration einer Variable versteht man das Bekanntmachen ihres Namens und ihres Typs. Nur externe globale Variablen können deklariert werden. Variablen anderer Arten von Speicherklassen können nur definiert werden. Der Unterschied der Definition gegenüber der Deklaration ist, dass Speicher für die Variable erzeugt wird.

Wie viele Gleitpunktdatentypen gibt es in C und wie unterscheiden sich diese?

Float und Double, Float mit 32-Bit und Double mit 64-Bit Speicherplatz.
Wobei der Datentyp double doppelt so genau ist als float.
Möglich ist auch long double der je nach Rechenarchitektur ≥ 64 Bit groß ist.

Was ist der Unterschied zwischen der Definition und Deklaration einer Funktion? Kann es eine Definition einer Funktion ohne eine Deklaration geben?

Die Definition ist die komplette Beschreibung der Funktion. Sie beinhaltet Funktionsrumpf (Typ, Name, Parameterliste) und Funktionsblock. die Deklaration ist eine Bekanntmachung einer Funktion an den Compiler, sie beinhaltet nur den Funktionsrumpf (wobei die Parameterliste weggelassen werden kann). Es ist möglich eine Funktion ohne Deklaration nur durch Definition zu verwenden, die def. muss dann aber vor dem ersten Aufruf der Funktion stehen. Und die Verwendung der Funktion ist ausschließlich in der Datei möglich.

Wie funktioniert die Parameterübergabe in C? Welche Arten der Parameterübergabe gibt es? Wie werden Felder üblicherweise an Funktionen übergeben? Wie kann man ein Feld als Kopie elegant an eine Funktion übergeben? (Antwort: Übergabe in einer Struktur)

Beim Aufruf der Funktion werden in runder Klammer die geforderten Variablen per Kommata getrennt übergeben. Variablen werden bei „call by value“ als Kopie des originalen Wertes übergeben. Bei Feldern würde eine Kopie oft sehr viel weiteren Speicher benötigen darum wird automatisch ein Zeiger auf das erste Element übergeben(„call by reference“ obwohl genau gesagt ein Zeiger mittels „call by value“ übergeben wird).

Kapitel 13 Felder

Wie werden Felder in C als Parameter einer Funktion übergeben?

Felder werden immer im Original übergeben (call by reference). Grund -> Speicherverbrauch und Geschwindigkeit

Was sind mehrdimensionale Felder und wie werden diese im Speicher repräsentiert?

Mehrdimensionale Felder sind in C eindimensionale Felder, bei denen jedes Feldelement wieder ein Feld ist. Solche Felder werden im Speicher hintereinander abgelegt. Jedes Feldelement hat dadurch zwei oder mehr Indizes zum ansprechen. Wobei der erste die Zeilen und der zweite die Spalten angibt. Vorsicht wird dieses Feld long feld[2][4] so angesprochen feld[0][4]=25; so ist dies falsch und das Element feld[1][0] wird gesetzt, da es an vierter Stelle im Speicher steht obwohl es zur zweiten Zeile gehört.

Welche Sortierverfahren kennen Sie?

Minimum-Suche, Bubble sort, Quicksort, Heap-sort

Minimumsuche bzw Maximumsuche (ist gleich) lasst euch nicht verwirren

Minimumsuche ist ein einfaches, langsames Verfahren. Suche in einer Folge von Zahlen das kleinste Element und tausche es gegen das Element an Position 0. Suche nun das zweit-kleinste Element und tausche es gegen das Element an Position 1. Minimumsuche wird zum Sortieren von Datensätzen in Dateien verwendet. Dreieckstausch

Wie arbeitet Bubblesort? Wie oft muss man dabei die zu sortierenden Daten durchlaufen? Was ist der unterschied zur "Maximumsuche" ?

Hier werden in einem Schritt immer zwei benachbarte Elemente miteinander verglichen, ist das linke größer als das rechte wird getauscht. Große Zahlen wandern von links nach rechts, da dieser Vorgang einem Bläschen gleicht, wird dies Bubblesort genannt. Im schlechtesten Fall muss das Verfahren N-1-mal wiederholt werden, wobei zu beachten ist, dass jeder Durchlauf kürzer als der vorangegangene dauert.

Kapitel 14 Zeiger

Kann eine Funktion einer Funktion übergeben werden?

Ja, mittels eines Zeigers auf eine Funktion (siehe nächste frage).

Wie definiert und verwendet man Zeiger auf Funktionen? Können diese auch innerhalb einer Funktion verwendet werden? Wie kann die Funktion aufgerufen werden?

Auch Funktionen haben eine Adresse. Man verwendet den Adressoperator um die Adresse einer Funktion zu erhalten. Definition von Zeiger auf Funktionen: Typ (* Funktionsname) (Parameterlist); Mit `fptr = &funkname` setzt man den Zeiger `fptr` auf die Funktion `Funkname`. Mit `x = fptr` (Variable) kann die Funktion aufgerufen werden. Der Rückgabewert ist wie bei normalen Funktionen. Einsatzgebiete für Zeiger auf Funktionen sind beispielsweise call back functions (Funktionen, die vom Betriebssystem beim Eintreten eines Ereignisses aufgerufen werden) oder Menüs. Bei den Menüs werden die einzelnen Menüeinträge von Strukturen beschrieben, welche einen Zeiger auf die gewählte Menüfunktion beinhaltet.

Was sind Zeiger?

Ein Zeiger ist eine Variable, die eine Adresse enthält. Will man das Objekt erreichen, auf das ein Zeiger zeigt, so muss der unäre Inhaltsoperator * dem Zeiger vorangestellt werden. -> der Zeiger wird dereferenziert.

Zeiger haben, wie reguläre Variablen, einen Datentyp. Es ist wichtig, Zeiger zu initialisieren. Nicht initialisierte Zeiger enthalten irgendeinen Wert. Soll eine Funktion mehr als einen Wert an den Aufrufer zurückgeben, müssen Zeiger als Parameter verwendet werden.

Zeiger-Feld Dualität erklären!

Software-technisch gesehen sind Zeiger und Felder zwei völlig verschiedene Konstrukte. Der Feldname kann in C wie ein Zeiger verwendet werden. Der Feldname steht für die Adresse, an der sich das Feld befindet. Auf Grund der "Zeiger-Feld-Dualität" kann der Index-Operator [] auch auf Zeiger, der Dereferenzierungsoperator auch auf Felder angewendet werden. Weiteres Beispiel für die Dualität: Wenn Felder als Parameter an eine Funktion übergeben werden. Denn auch wenn die Parameter einer Funktion als Felder definiert werden, sie werden als Zeiger verwendet.

Was sind zeiger auf zeiger und wann verwendet man sie? Er hat mich nach einer anderen Verwendung auch noch gefragt.

Zeiger können auf einen beliebigen Datentyp zeigen. so können zeiger auch auf weitere zeiger gesetzt werden. man verwendet Zeiger auf Zeiger um zwei Zeiger zu tauschen (swap zeiger -> Parameter sind zeiger auf zeiger, Dreieckstausch)

Kapitel 15 Zeichenketten

Was ist ein String?

Zeichenketten sind nullterminierte Folgen von Zeichen, die in Feldern von Zeichen abgespeichert werden. Zeichenketten sind in C nicht als ein unabhängiger Datentyp implementiert. Bei Zeichenketten unter Anführungszeichen ist das Null-Byte automatisch angehängt.

Kapitel 16 Abgeleitete Datentypen

Was sind variante Strukturen? warum legt man nicht einfach beide Auswahlmöglichkeiten explizit an?

Variante Strukturen haben große Ähnlichkeit mit regulären Strukturen. die Attribute der varianten Strukturen belegen denselben Speicherbereich, sie liegen quasi übereinander. Variante Strukturen werden mit dem Schlüsselwort union. definiert. Die Länge einer varianten Struktur entspricht der Länge des größten Attributs.

Was ist eine rekursive Struktur?

Rekursive Strukturen sind Strukturen die Attribute enthalten die auf Strukturen desselben Typs verweisen.

```
struct Eintrag_s  
{ struct Eintrag_s *naechster;  
  long wert;  
}
```

Nächster ist ein Zeiger und zeigt auf ein weiteres Objekt dieser Struktur. Dies ist möglich da die Länge eines Zeigers(32bit, 4 Byte) bekannt ist.

Verwendung -> dynamische Speicherverwaltung

Erkläre typedef!

Mit typedef können neue Typnamen erzeugt werden, wobei die neu erzeugten Typnamen zu der Bezeichnung für die sie stehen vollkommen äquivalent sind. Die Anwendung von Typedef dient der Lesbarkeit eines Quellcodes, sowie der Verkürzten Schreibweise für Verbunddatentypen. (Man erspart sich das „struct“). Mit typedef wird kein neuer Datentyp definiert. Es wird ein weiterer Name für einen bestehenden Typ geschaffen.

Kapitel 17 Dateien

Kapitel 18 Rekursive Funktionen

Was sind rekursive Funktionen? Was passiert wenn die Abbruchbedingung erfüllt ist?

Eine Funktion die sich selbst aufruft nennt man rekursiv. Jeder Durchlauf der Funktion trägt zur Gesamtlösung bei. Rekursive Funktionen werden in C mittels eines Stapels realisiert wobei jede Rekursionsebene ihr eigenes Variablenset am Stapel bekommt. Durch jeden Selbstaufwurf wird der Stapel größer, ist die Abbruchbedingung erfüllt, wird also eine Funktion beendet, so wird die zugehörige Rekursionsebene zerstört und zur vorherigen zurückgekehrt. Beispiele für die Verwendung rekursiver Algorithmen: Quicksort und das Traversieren von Bäumen

Wie funktioniert Quicksort?

Quicksort ist ein sehr schnelles Sortierverfahren und arbeitet besonders effizient bei unsortierten Datenmengen. Vorteile: geringer Speicherverbrauch und relativ einfach zu implementieren. In jedem Schritt wird ein bestimmtes Element an seine korrekte Position gebracht. In einem Schritt wird gesucht und eingeordnet.

Zum Sortieren vollkommen unsortierter Datenmengen in einem Feld, arbeitet rekursiv. Die Bedingung, dass links vom neu positionierten Element nur kleinere und rechts vom neu positionierten Element nur größere Elemente vorkommen, muss gewährleistet werden.

Es wird ein beliebiges Element ausgewählt, beispielsweise das ganz rechte, es wird weiter eine linke und eine rechte Grenze auf die restlichen Elemente gelegt innerhalb denen weitergearbeitet wird. Von der linken Grenze aus wird ein Element, welches größer ist als das beliebig gewählte, mit einem von der rechten Grenze aus gesuchten Element, welches kleiner als das beliebig gewählte Element ist, ausgetauscht. Danach können die Grenzen enger zueinander rücken und der Vorgang wiederholt werden bis die Grenzen aneinander liegen. An diesem Punkt kann das beliebige Element an die Stelle gerückt werden an der es links von sich nur kleinere und rechts von sich nur größere Elemente vorfindet. Nun kann Quicksort für das linke und das rechte Teilfeld aufgerufen werden.

Kapitel 19 Datenstrukturen

Was ist eine einfach verkettete Liste und wie ist diese aufgebaut?

In Behältern werden Daten gespeichert und verwaltet. Behälter: Felder, Listen, Bäume, Heaps und Hashes. Unter einer Liste versteht man eine sequenzielle Aneinanderreihung von Elementen. Die Reihenfolge der Elemente wird explizit festgehalten. Eine Liste besteht aus Knoten. Ein Knoten enthält zwei Informationen: das Element und die Verkettung). Der Anfang der Liste - auch Kopf genannt- wird separat gespeichert.

In C versteht man unter verkettete Liste eine sequenzielle Aneinanderreihung von Elementen. Die Reihenfolge wird in der Liste explizit festgehalten.

Ein sog. Knoten der Liste wird durch einen Verbunddatentyp realisiert welcher den Wert/Buchstaben und einen Verweis auf das nächste Listenelement beinhaltet, der Kopf verweist auf das erste Listenelement. Ein Feld von Knoten muss nur mehr initialisiert und die Verkettungen getroffen werden. Vorsicht: der letzte Knoten muss im Verweis gekennzeichnet sein um das Listenende erkennbar zu machen.

Wie funktioniert das "Einfügen" in Listen?

Will man ein Element einfügen muss die Liste an der gewünschten Position aufgebrochen, und auf den neuen Knoten verbogen werden. Die Verknüpfung des neuen Elements wird auf die restliche Liste gesetzt. An erster oder letzter Stelle einfügen ist, solange Feldgrenze nicht überschritten wird, leicht. Soll ein Element mitten in die Liste eingefügt werden muss an der richtigen Stelle die Verknüpfung des Vorgängers des neuen Elements auf das neue Element gesetzt, und der Nachfolger des Vorgängers des neuen Elements im neuen Element festgehalten werden.

Was ist eine Ringliste?

Eine Liste bei der der Endknoten einen Verweis an den Anfang der Liste speichert.

Erklären Sie die Datenstruktur Schlange.

Eine Schlange arbeitet nach dem Prinzip FIFO. Die Methoden die für eine Schlange angeboten werden sind Put und Get (Elemente). Schlangen werden vorzugsweise mit Listen implementiert. Zirkularpuffer werden mit Feldern implementiert.

Was ist ein Zirkularpuffer ?

Der Zirkularpuffer ist eine Implementierung der Schlange und arbeitet daher auch nach dem Prinzip des FIFO. Zirkularpuffer werden in Feldern implementiert. die Einfüge- und Entnahmeposition sind durch Indizes im Feld gegeben. Nach dem/der Einfügen/entnahme eines Elements wird der Index der Einfügeposition/Entnahmeposition erhöht. Es verschieben sich daher die Positionen im Feld. Wird die Feldlänge überschritten, so wird der jeweilige Index auf den Feldanfang gesetzt. Zirkularpuffer werden gerne bei der asynchronen Kommunikation zweier Prozesse verwendet. zwei Problemsituationen: underflow, overflow.

Erkläre Bäume

Ein Baum ist ein Behälter. Es gibt Bäume zur Speicherung von Daten, Bäume zum Suchen und Sortieren von Daten, Bäume für Syntaxanalyseprobleme, geometrische oder mathematische Probleme oder für die Datenkomprimierung. Jeder Knoten hat nicht nur einen sondern auch mehrere Nachfolger. (Baum zeichnen) Blätter, Wurzel, Pfad, Kante, Vorgänger, Nachfolger, innerer Knoten, äußerer Knoten, Wald. Werden Elemente nach einer bestimmten Ordnung in den Baum eingefügt, so wird ein so genannter Schlüssel benötigt.

Was ist ein binärer Baum?

Unter binären Bäumen versteht man einen Bäume, die deren Knoten genau zwei Nachfolger haben. Bei einem vollständigen Baum sind die Blätter von links nach rechts angefügt.

Binäre Suchbäume

Bei einem binären Suchbaum erfüllt jeder Knoten eine besondere Bedingung: alle Knoten, die sich im linken Unterbaum eines Knotens befinden, haben kleinere Schlüssel. Ein binärer Baum bei dem alle Schlüssel unterschiedlich sind kann in einen vollständigen Baum umgewandelt werden.

was versteht man unter Traversierung?

Die Traversierung beschreibt wie ein Baum oder ein Feld durchlaufen wird. Es gibt verschiedene Methoden wie ein Baum durchlaufen wird: Level-Order, Preorder, Inorder, Postorder-Traversierung. Die Methode Pre- In und Postorder sind ähnlich. Der Unterschied ist die Reihenfolge in der die Knoten durchlaufen werden. Preorder -> B-A-C, Inorder -> A-B-C, Postorder -> A-C-B Level-Order-Methode kommt bei Heap-Strukturen zum Einsatz. Dabei werden die Elemente zeilenweise von links nach rechts durchlaufen.

Suchen eines Elementes in einem binären Suchbaum

Soll ein Element in einem Baum gesucht werden, so muss durch den Baum traversiert werden. Kleinere Schlüssel links, größere rechts. Bei einer erfolgreichen Suche -> $\log_2(N)-1$ Vergleiche. erfolglose Suche -> $\log_2(N)$ Vergleiche.

Einfügen eines neuen Elementes in einen binären Suchbaum

Um einen Knoten in einen binären Suchbaum einzufügen, wird das Element zunächst im Baum erfolglos gesucht. Die Suche wird abgebrochen, wenn ein Endknoten erreicht worden ist.

Löschen eines Elementes in einem binären Suchbaum

Ein Löschen eines Elementes kommt einer Beschädigung der Struktur des Baumes gleich, die repariert werden muss. Es muss ein anderer Knoten, mit dem nächstgrößten oder nächstkleinsten Schlüssel gesucht werden, den Platz vom gelöscht Knoten einnehmen.

Was ist ein Heap und für welche Operationen ist er gut geeignet?

Wie können die Indizes der Nachfolger eines Elements berechnet werden?

Ein Heap ist eine besondere Art eines binären Baumes, der als Feld abgespeichert wird. Jeder Knoten muss dabei die so genannte heap-Bedingung erfüllen: Der Schlüssel jedes Knotens ist größer als der seiner Nachfolger. Ein heap ist ein vollständiger binärer Baum. Jedem Knoten wird ein fortlaufender Index zeilenweise zugewiesen. Diese Indizes entsprechen den Feldindizes, wobei das Element mit Index 0 nicht existiert. Vorteile bei der Implementierung als Feld: heaps speichern keine expliziten Verknüpfungen -> feststellen des Nachfolgers durch einfache arithmetische Operationen. Nachfolger Index i -> $2i$ und $2i + 1$, Index des Vorgängers -> Ganzzahldivision $i/2$. Nachteil: starre Struktur. Methoden: wenn heaps als Prioritätswarteschlangen verwendet werden: Einfügen eines neuen Elements, Traversieren über einen heap (meist level-order-methode oder Preorder-Traversierung), entfernen eines Elements. Nach dem Einfügen oder Löschen muss die heap-Struktur wieder repariert werden. UpHeap arbeitet sich von unten nach oben durch den heap - sie arbeitet bottom up. Ein neues Element wird an die letzte Position eingereiht. -> heap Bedingung ist verletzt. Der Schlüssel des neuen Elementes wird immer mit seinem Vorgänger verglichen. DownHeap: arbeitet sich von oben nach unten durch den heap - sie arbeitet top down. Diese Methode wird vor allem beim Ersetzen von Elementen angewendet.

Entfernen von Elementen

Das zu entfernende Element wird mit dem letzten heap-Element getauscht. Danach wird die Länge des heaps um 1 reduziert, wodurch das Element entfernt wird. Danach wendet man DownHeap an um die Struktur wieder zu ordnen.

Wie funktioniert Heap- Sort? Wie kommt man beim Heap auf einen Nachfolger?

Heaps ermöglichen das effiziente Sortieren von Feldern in ungefähr $N \log_2(N)$ Schritten. Algorithmus: Entferne die Wurzel aus dem heap. Wiederhole den Vorgang solange bis der heap leer ist. Die Elemente liegen dann in sortierter Reihenfolge im Feld vor. Wurzel löschen dann DownHeap anwenden.

Erkläre Hash

In einem Feld werden die Datensätze gespeichert und mit einem generierten Index gesucht. Die hashFunktion transformiert einen Schlüssel in eine ganze Zahl. Die Indizes verschiedener Schlüssel können gleich sein -> Kollisionsbeseitigung Lösung: Datensätze mit gleichem Schlüssel in einer verketteten Liste zusammenfassen

Wie löscht man ein Element aus einem Hash?

Element einfach als ungültig markieren, bei Liste die Zeiger auf den Nachfolger vorher setzen.

Was ist ein Hash und für welche Operationen ist er gut geeignet? Wie groß sollte die Hash- Länge gewählt werden?

Hashes speichern Daten in einem Feld ab, wobei der Index des Feldelements mit einer Hash- Funktion ermittelt wird. Die Qualität des Hashes hängt aber stark von der verwendeten Hash- Funktion, der Feldlänge und den Schlüsseln der zu speichernden Daten ab. Hashes eignen sich hervorragend zum suchen von Daten und somit zur Datenerhaltung. Sie speichern jedoch keine Reihenfolgen, Hierarchien oder sonstige Abhängigkeiten zwischen Daten ab. Kenne ich die Datenmenge M so sollte man die Hash- Länge stets größer wählen ($N \geq 3M$, N als Primzahl von Vorteil).

Was wird aus dem Hash, wenn man bei getrennter Verknüpfung Hash Länge 1 wählt? Antwort: verkettete Liste

Kapitel 20 Dynamischer Speicher

Erklären Sie die dynamische Speicherverwaltung in C. welche funktionen werden angeboten und

wie funktionieren sie?

Aus einem Programm wird Speicher einer bestimmten Größe von der Speicherverwaltung angefordert, im Programm verwendet und nach der Verwendung an die Speicherverwaltung wieder "retourniert". Um das Zerstückeln des Speichers durch viele kleine Blöcke zu vermeiden, runden viele Betriebssysteme die geforderte Länge auf beispielsweise durch 8 teilbare Längen auf. (wenn man 2 Byte anfordert bekommt man 8 Byte) Funktionen:

malloc : Als einziger Parameter wird die Länge des Speichers in Byte verlangt. der rückgabewert der Funktion ist die Adresse des Speicherblocks. konnte kein Speicherplatz alloziert werden, so wird der wert 0 zurückgegeben. Vorsicht! der Datentyp wird nicht geprüft.

calloc : Der Speicher wird mit 0 vorinitialisiert. calloc hat 2 Parameter: erster Parameter wird die Anzahl der Elemente des zu allozierenden Feldes und als zweiter Parameter die Größe eines Elementes in Byte erwartet.

der neue Speicherblock kann an einer anderen Adresse liegen. Vorsicht wenn man mit Zeigern arbeitet.

realloc : Mit dieser Funktion könne Speicherblöcke in ihrer Größe verändert werden. 2 Parameter:

realloc(alterBlock, neue länge)

free : wenn dynamischer Speicher nicht mehr benötigt wird, sollte er wieder freigegeben werden. free(adresse). Speicher darf nicht zweimal freigegeben werden. deswegen sollte man per Konvention den Zeiger auf 0 setzen.

was sind memory-leaks ?

Leaks sind ungenutzter Speicher. dieser Speicher kann nicht mehr verwendet oder gelöscht werden, da die Anfangsadresse nicht mehr bekannt ist. Folgen sind unnötiger Speicherverbrauch eines Programms oder sogar Blockade des Rechners. Durch folgende Fehler können Leaks entstehen: Verlust des Zeigers durch Überschreiben des Zeigers, Verlust des Speichers durch Verlust des Zeigers, Verlust des Speichers durch Rücksprung und Verlust des Speichers bei Rückgabe.

Kapitel 21 Numerik

Welche Fehlerarten gibt es? (Numerik)

Modellfehler, Datenfehler, Verfahrensfehler und Rundungsfehler

Was sind Modellfehler?

Bei jeder Modellbildung wird eine Abstraktion der Realität durchgeführt. Dabei werden bekannte und unbekannte Größen vernachlässigt und Vereinfachungen der tatsächlichen Verhältnisse vorgenommen. Der Dadurch entstandene Fehler wird Modellfehler genannt. Der Beitrag zum Gesamtfehler kann aber nur von den bekannten Größen nicht aber von den unbekanntem, nicht beachtetem Größen abgeschätzt werden.

Was sind Datenfehler?

Datenfehler entstehen durch Ungenauigkeiten bei der Erfassung von Daten. Diese Messfehler lassen sich mittels Kondition abschätzen.

Was sind Verfahrensfehler?

Oft werden Probleme mit numerischen Näherungsverfahren die sich iterativ an die Lösung herantasten gelöst. Wenn eine Lösung hinreichend genau ist wird das Verfahren abgebrochen, so entsteht der Abbruchfehler.

So genannte Diskretisierungsfehler entstehen beim Übergang von kontinuierlichen auf diskrete Systeme. Beispielsweise wird die Integralbildung numerisch durch eine Summation durchgeführt, bei der gegenüber dem Integral ein Fehler auftritt.

Durch hohen Aufwand erreicht man kleinen Verfahrensfehler, durch Abbruchfehler, aber nicht kleiner als

Darstellungsgenauigkeit des eingebauten Datentyps.

Was sind Rundungsfehler?

Da Punktzahlen auf Computern nur mit bestimmter Genauigkeit dargestellt werden können muss gerundet werden. Dadurch entsteht der Rundungsfehler. Der Unterschied zwischen dem mathematisch exakten und dem gerundeten Ergebnis heißt Rundungsfehler oder Rechenfehlern.

Was sagt die Konditionszahl aus?

sie gibt die Auswirkung einer Änderung der Eingangsdaten D auf die Lösung x an. diese zahl wird abgeschätzt. ein System, das unempfindlich gegenüber Änderungen der Eingangsdaten ist, wird als gut konditioniert genannt(große Konditionszahl),

kleinste möglich zahl?

Die header-Datei limits.h gibt Auskunft über die Grenzen ganzzahliger Zahlentypen. Konstanten: LONG_MAX größte positive Zahl, LONG_MIN kleinste negative Zahl. die header-Datei float.h gibt die Schranken für die Datentypen float, double und long double an. DBL_MAX größte positive Zahl, DBL_MIN kleinste normalisiert positive Zahl > 0, DBL_EPSILON Differenz zwischen 1 und der kleinsten Zahl größer als 1

Was passiert wenn 2 double werte addiert werden?

Bei der Summation zweier Zahlen werden die Mantissen addiert. Dazu müssen die Exponenten gleich sein. Ist die Zahl um den Faktor 2^{52} größer wird 0 addiert. In der Computernumerik gelten weder das Assoziativgesetz noch das Distributivgesetz.

Welche Probleme können bei der Summation von Punktzahlen auftreten?

Auslöschungseffekt bei Summation annähernd gleicher Werte. Unterscheiden sich zwei errechnete Werte ausschließlich in den hinteren Kommastellen, so bleiben diese unbedeutsamen Beträge bei der Subtraktion der Werte übrig. Die Störungen wirken sich dann auf den relativen Fehler aus. Auslöschung ist die häufigste Ursache von schlechter Kondition.

Kapitel 22 Fehlerbehandlung

Welche Fehlerarten gibt es? (Fehlerbehandlung)

Modellfehler, Datenfehler, Verletzte Vorbedingung, Anwenderfehler, Fehlerhafter Programmaufruf, Numerische Fehler, Speicherproblem, Fehlerhafter Speicherzugriff, Fehlerhafte Module und Dead lock

Welche Möglichkeiten bietet C zur Fehlerbehandlung und wie funktionieren diese?

Verschiedene Programmiersprachen ermöglichen verschiedene Arten der Fehlerbehandlung. Zum Beispiel -> "Ausnahmen" (exceptions) bei C++. In C haben sich so genannte "Fehlercodes" durchgesetzt. Es empfiehlt sich Fehlercodes mit Aufzählungstypen zu realisieren. Der Präfix ERR deutet an, dass es sich um einen Fehlercode handelt. Die C-Standard-Bibliothek bietet einige Möglichkeiten zur einfachen Fehlererkennung und Fehlerbehandlung. die globale Variable errno vom Typ int ist in der Standard.Header-Datei errno.h deklariert. Zum Programmstart wird der Wert automatisch auf 0 gesetzt. Der Wert wird danach nie auf 0 gesetzt. Das sollte man dann vor dem Aufruf der zu untersuchenden Funktion machen. ENOENT (Präprozessorkonstante)

Was macht die Funktion "exit" und wie verwendet man sie?

Hilfreich für die Programmentwicklung, mit exit wird das Programm unmittelbar beendet, egal wo sich das Programm gerade befindet. Die Funktion exit ist in der header-Datei stdlib.h deklariert und erwartet als

einigen Parameter einen Fehlerwert. so wie die main Funktion.
Aufruf der Funktion: `exit(-1);` //ein Parameter/Fehlerwert.

Was macht die Funktion "atexit" und wie verwendet man sie?

```
#include <stdlib.h>
```

Mit der Funktion `atexit` können mehrere Funktionen registriert werden, die beim regulären Programmende aufgerufen werden. Ein Programmende ist dann regulär, wenn aus der Funktion `main` rückgesprungen, oder `exit(-1)`; aufgerufen wird. Werden mehrere Funktionen registriert, erfolgt der Aufruf der Funktion in der umgekehrten Reihenfolge in ihrer Registrierung. Einziger Parameter -> Zeiger auf eine Funktion.

```
void dasEnde()  
{ printf(„dasWars!“);  
}  
main()  
{blabla...  
  atexit(&dasEnde);  
}
```

Die Funktion die aufgerufen werden soll darf keinen Rückgabewert und keine Parameter haben.

Was macht die Funktion "assert" und wie verwendet man sie?

```
#include <assert.h>
```

```
BSP.: datei=fopen(„beispiel.txt“,“r“);
```

`assert(datei);` // ist der Parameter (hier `datei`) 0 so wird eine Fehlermeldung ausgegeben und das Programm mittels `abort` beendet. Die Fehlermeldung beinhaltet den Namen der Quelltextdatei und die Zeilennummer, in der der Abbruch stattgefunden hat. `Abort`, im Unterscheid zu `exit`, beendet das Programm durch einen Abbruch und sendet das Signal `SIGABRT` an den eigenen Prozess.

Wie funktioniert die Signalbehandlung in C? Was ist ein Signal? Welche Signale kennen Sie?

Prozesse können sogenannte Signale vom Betriebssystem erhalten. Die header-Datei `signal.h` enthält eine vollständige Liste aller vorhandenen Signale. Empfängt ein Prozess ein Signal, wird der Programmablauf unterbrochen und eine Funktion zur Behandlung des Signals aufgerufen, ein so genannter `signal handler`. Die Funktion `signal` ist in der header-Datei `signal.h` deklariert und erwartet als erstes Argument das Signal und als zweites Argument den Zeiger auf die Signalbehandlungsfunktion, die beim Aufrufen des Signals aufzurufen ist. Eine Signalbehandlungsfunktion hat einen Parameter vom Typ `int`: Das Signal, das zum Aufruf der Signalbehandlungsfunktion geführt hat. Eine Sbf hat keinen Rückgabewert. Signale: `SIGINT` Unterbrechung (interrupt): Dieses Signal wird beim Drücken der Tastenkombination `Strg-C` an den Prozess gesandt. `SIGABRT` Abbruch (`abort`): Dieses Signal wird von den Funktionen `abort` und `assert` verwendet. `SIGTERM` terminieren (`terminate`): Dieses Signal wird beim Herunterfahren des Systems an jeden Prozess gesandt. empfohlen. `SIGILL` Illegaler Maschinenbefehl (`illegal instruction`): tritt auf wenn ein ungültiger Maschinenbefehl geladen wird. `SIGFPE` Gleitpunkt-Fehler (`floating point exception`) tritt auf, wenn eine ungültige arithmetische Operation durchgeführt wird. `SIGSEGV` Unerlaubter Zugriff auf Speicher (`segmentation violation`). Dieses Signal wird an einen Prozess gesandt, wenn er versucht, auf einen Speicher zuzugreifen, der nicht zum Prozess gehört oder wenn eine illegale Adresse auftritt.

Erkläre die Funktion `abort()`?

Die Funktion `abort` beendet das Programm durch Abbruch. Dabei wird das Signal `SIGABRT` an den Prozess gesandt, wodurch die Signalbehandlung für dieses Signal ausgelöst wird. Die voreingestellte Signalbehandlung bricht das Programm ab. Die Funktion `abort` ist in der header-Datei `stdlib.h` deklariert und hat keine Argumente. Die Funktion `abort` wird bspweise in der Funktion `assert` verwendet.

Wie funktioniert das Sortierverfahren "Maximumsuche"?

Wie funktioniert `printf`?

wie funktioniert `scanf`?

Referenzbeispiele!!!
Wie kopiert/übergibt man felder?