

Anmerkungen

- Sichern Sie ihre Berechnungen, so weit wie möglich gegen fehlerhafte Eingaben ab.
- Verwenden Sie sprechende Variablennamen und formatieren Sie Ihren Programmtext ordentlich.
- Erstellen Sie Inputprompts und Ausgaben, sodass das Programm für BenutzerInnen angenehm zu bedienen ist.
- Schreiben Sie Ihre Programm so, dass mehrere Berechnungen ausgeführt werden können, ohne das Programm immer wieder neu starten zu müssen.
- Testen Sie Ihre Programme und unterziehen Sie die Ergebnisse einer Prüfung (Schätzung des erwarteten Ergebnisses, bzw. im Notfall mit dem Taschenrechner prüfen).
- Mit * markierte Beispiele sind etwas anspruchsvoller

Aufgabe 1 *

Lösen Sie die Zusatzaufgaben aus der Übungseinheit so vollständig wie möglich.

Aufgabe 2

Lesen Sie beliebig viele (max. 50) Zahlen (Abschluss mit dem Wert 0) ein, speichern Sie diese in einem Vektor und geben Sie Mittelwert, Median und die Standardabweichung der Zahlen aus. (Bei der Ermittlung des Medians können Sie davon ausgehen, dass die Zahlen schon sortiert sind, vom Benutzer also entweder in aufsteigender oder in fallender Reihenfolge eingegeben werden.)

Aufgabe 3

Bestimmen Sie alle Primzahlen bis zu einem gegebenen n mit Hilfe des Siebs des Eratosthenes.

Aufgabe 4

Lesen Sie 2 2×2 Matrizen ein und geben Sie das Produkt der beiden Matrizen aus. Legen Sie das Programm so aus, dass der Benutzer beliebig viele Matrizenmultiplikationen ausführen kann, ohne das Programm mehrmals zu starten. Verwenden Sie in Ihrem Programm zur Speicherung der Matrizen maximal 3 Variable.

Aufgabe 5

Lesen Sie maximal 50 Messpunkte (x_i/y_i) ein und berechnen Sie die Koeffizienten a und b in der Gleichung der Regressionsgeraden $y=ax+b$. Geben Sie außerdem den Punkt aus, dessen y Koordinate am stärksten von der berechneten Regressionsgeraden abweicht.

Aufgabe 6

Bei einer Wahl muss sichergestellt werden, dass jeder Wahlberechtigte (es sind derer maximal 90) nur einmal seine Stimme abgibt. Für diesen Zweck prüft man die Sozialversicherungsnummer und merkt sich, wer schon gewählt hat. Schreiben Sie ein Programm, das beliebig viele Sozialversicherungsnummern einliest und eine Warnung ausgibt, sobald eine Nummer zum zweiten Mal eingegeben wird.

Überlegen Sie, ob für diese Anforderungen der Einsatz einer Hash-Tabelle angebracht ist und verwenden Sie gegebenenfalls eine Hash-Tabelle mit linear probing

Aufgabe 7

Schreiben Sie ein Programm, das Tipps im Lotto 6 aus 45 verwaltet. Lesen Sie maximal 50 Tipps ein (ein Tipp besteht aus jeweils 6 Zahlen. Beenden der Eingabe mit 0) und danach das Ziehungsergebnis (ebenfalls 6 Zahlen). Geben Sie aus wieviele Sechser in den Tipps vorkommen.

Anmerkung: Gehen Sie der Einfachheit halber davon aus, daß die einzelnen Zahlen in den Tipps und im Ziehungsergebnis aufsteigend sortiert eingegeben werden.

Aufgabe 8

Lesen Sie maximal 50 Zahlen (Abschluß mit 0) in einen Vektor V ein, sowie einen Wert x. Teilen Sie den Inhalt des Vektors V so auf zwei weitere Vektoren Vg und Vk auf, dass alle Werte, die größer als x sind, in Vg liegen und alle anderen in Vk.

Geben Sie dann abwechselnd je eine Zahl aus Vg und eine aus Vk aus, solange bis alle Zahlen ausgegeben sind.

z.B. Eingabe: 3 14 31 7 29 10 18 15 19 0 x=10

Vg enthält dann 14 31 29 18 15 19 und Vk 3 7 10

Ausgabe: 14 3 31 7 29 10 18 15 19

Aufgabe 9

Lesen Sie maximal 50 Zeichenketten ein (Abschluss durch Eingabe von "Stop") sowie einen Wert n.

Geben Sie dann alle Zeichenketten aus, die maximal n Zeichen lang sind.

Anmerkung: Beachten Sie, dass beim Einlesen eine Zeichenkette durch ein Leerzeichen beendet wird.

z.B.

Eingabe: C++ ist ganz schön kompliziert so im Allgemeinen Stop n=4

Ausgabe: C++ ist ganz so im

Aufgabe 10

Schreiben Sie ein Programm zur Ermittlung des zu entrichtenden Steuerbetrags aus dem Gewinn unter Berücksichtigung der Steuerprogression. Verwenden Sie einen Vektor, der mit den entsprechenden

Bereichsgrenzen initialisiert wird, sowie einen zweiten Vektor, der mit den zugehörigen Steuersätzen initialisiert wird. Die aktuellen Werte für Österreich können Sie aus der Tabelle entnehmen und im Programm als Konstante vorgeben (Die Tabelle muss also nicht eingelesen werden.) Das Programm soll einen Gewinnbetrag einlesen und die entsprechende Steuer berechnen und ausgeben.

von	bis	Prozentsatz
0	11000	0
11000.01	18000	25
18000.01	31000	35
31000.01	60000	42
60000.01	90000	48
90000.01	1000000	50
1000000.01		55

Aufgabe 11 *

Bei einem Schiwettkampf mit maximal 30 Teilnehmern werden die erreichten Zeiten im Format min:sec:tausendstel eingegeben. Schreiben Sie ein Programm, das die Anzahl der Teilnehmer und die erreichten Zeiten einliest. Danach soll die Bestzeit ermittelt werden. Ausgegeben wird die Startnummer des Siegers, die Bestzeit, sowie eine Tabelle aller Teilnehmer in der die Zeitdifferenz zum Sieger angeführt wird. Diese Tabelle wird nach Startnummern sortiert ausgegeben, also nicht nach erreichter Position. z.B.:

Eingabe:

5

1:10:22

1:5:22

1:7:40

1:8:27

1:6:23

Ausgabe:

Sieger ist die Startnummer 2 mit 1:5:22

1 5

2 0

3 2.18

4 3.05

5 1.01

Aufgabe 12 *

Schreiben Sie eine einfache Version des Tic-Tac-Toe Spiels. Dabei spielt der Benutzer gegen den Computer. Nach jedem Zug soll der aktuelle Zustand des Spielbretts am Bildschirm ausgegeben werden.

(Der Computer muss sich dabei nicht besonders schlau anstellen. Es genügt, wenn er einfach irgendeines der freien Felder besetzt, sobald er am Zug ist. Es muss auch nicht automatisch erkannt werden, wenn die Partie zu Ende ist.)

Aufgabe 13

Schreiben Sie ein Programm, das mit dem Gaußschen Eliminationsverfahren ein quadratisches lineares Gleichungssystem (n Gleichungen mit n Unbekannten) löst.

Eingegeben wird die Anzahl der Variablen, sowie die Koeffizienten und rechten Seiten der Gleichungen. Ist das Gleichungssystem eindeutig lösbar, so soll die Lösung ausgegeben werden, andernfalls eine Fehlermeldung

Aufgabe 14

Eine quadratische Matrix heißt schwach diagonaldominant (streng diagonaldominant), falls der Absolutbetrag der Einträge in der Diagonale jeweils größer gleich (größer) ist, als die Summe der Beträge der jeweiligen anderen Elemente in der gleichen Zeile.

Schreiben Sie ein Programm, das die Dimension einer quadratischen Matrix und die Matrixelemente einliest und prüft, ob es sich um eine streng, schwach oder nicht diagonaldominante Matrix handelt.

Aufgabe 15

Schreiben Sie ein Programm, mit dem man "Schiffe versenken" spielen kann. Es soll möglich sein, die Position der Schiffe manuell festzulegen (besonders Eifrige können auch versuchen, die Positionierung automatisch durchzuführen, etwa unter Verwendung von Zufallszahlen; dazu können Sie nach den Funktionen `rand` und `randomize` googeln). Anschließend kann durch Angabe der Koordinaten auf ein Feld geschossen werden. Der Computer soll die üblichen Meldungen ("Wasser", "Treffer", "Versenkt") liefern und erkennen, wenn das Spiel beendet (die gesamte Flotte vernichtet) ist.

Aufgabe 16

Beim 8-Damen-Problem müssen auf einem Schachbrett 8 Damen so positioniert werden, dass sie sich gegenseitig nicht bedrohen.

Schreiben Sie ein Programm zur Unterstützung bei der Suche nach einer Lösung. Es soll möglich sein, durch Eingabe der Feldkoordinaten in der üblichen Schachnotation Damen auf dem Feld zu positionieren. Falls die gewählte Position bedroht ist, soll das Programm eine Meldung ausgeben. Andernfalls wird eine Dame an diese Position gesetzt. Die Eingabe kann so lange wiederholt werden, bis 8 Damen positioniert wurden.

Aufgabe 17

Schreiben Sie ein Programm, das alle möglichen Lösungen für das 8-Damen-Problem (siehe Aufgabe 16) ermittelt und ausgibt.

Aufgabe 18

Wie Aufgabe 15, allerdings soll das Spielfeld dreidimensional sein und die Größe des Spielfelds darf vor Spielbeginn frei gewählt werden.

Aufgabe 19

Wie Aufgabe 16, allerdings für das n -Damen-Problem. Das heißt, es müssen n Damen auf einem $n \times n$ Feld positioniert werden.

Aufgabe 20 *

Conways 'Game of Life' besteht aus einem unendlichen Raster von quadratischen Zellen. Jede dieser Zellen kann einen von zwei Zuständen annehmen: "lebendig" oder "tot". Aus einer gegebenen Konfiguration kann die "nächste Generation" nach folgenden Regeln gebildet werden:

1. Jede lebende Zelle mit weniger als 2 lebenden Nachbarzellen stirbt an Unterbevölkerung
2. Jede lebende Zelle mit 2 oder 3 lebenden Nachbarzellen überlebt.
3. Jede lebende Zelle mit mehr als 3 lebenden Nachbarzellen stirbt an Überbevölkerung
4. Eine tote Zelle mit genau 3 lebenden Nachbarzellen wird zu einer lebenden Zelle (Vermehrung)

Die Regeln sind auf alle Zellen einer Generation gleichzeitig anzuwenden, um die nächste Generation zu erhalten.

Schreiben Sie ein Computerprogramm, das ein endliche Version von 'Game of Life' realisiert. Die Dimension des $n \times m$ Spielfeldes soll zu Beginn festgelegt werden. Danach können die lebenden Zellen der ersten Generation eingegeben werden und der Ablauf wird gestartet. Das Programm gibt dann Schritt für Schritt die jeweiligen Nachfolgenerationen aus. Gehen Sie für die Berechnung davon aus, dass rund um das Spielfeld weitere Nachbarzellen existieren, die alle tot sind.

Aufgabe 21 *

Implementieren Sie Conways 'Game of Life' wie in Aufgabe 20, gehen Sie aber davon aus, dass die Spielfelder am Rand zyklisch fortgesetzt sind. Das heißt, wenn das Spielfeld z.B. n Zeilen hat, dann ist die $n+1$. Zeile gleich der 1. Zeile. Analog für Spalten.

Aufgabe 22 *

Schreiben Sie ein Programm, das beliebig viele Zahlen einliest und fallend sortiert wieder ausgibt.

Verwenden Sie zum Sortieren folgenden rekursiven Algorithmus: Zunächst wird das Maximum gesucht und an die erste Stelle der Liste gesetzt. Dann wird dieser Vorgang mit der restlichen (um ein Element kürzeren) Liste wiederholt, so lange bis die Länge der Liste 1 ist.

Für die Programmierung nehmen Sie am besten einen Vektor für die Liste und reichen diesen als Referenz von einem Funktionsaufruf zum nächsten, ohne die Länge wirklich zu verringern. Verwenden Sie einen zweiten Parameter, der die Startposition im Vektor angibt und sich mit jedem Funktionsaufruf erhöht.