Praktikum C-Programmierung 2026

Vorstellung Abschlussprojekte

Jannek Squar

2026-06-17

Scientific Computing
Universität Hamburg

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Outline

Table of Contents

Dateistruktur

- Modulare Programmierung
- Unterteilung in sinnvolle Funktionen
  - „Funktionsinhalt sollte auf Bildschirm passen“
- Main-Funktion sollte nur Funktionsaufrufe enthalten
- Funktionalität in eigene shared Library auslagern
- Keine intensive Nutzung von existierenden Libs, insbesondere für String-Manipulation oder File I/O

Programmgestaltung

- Eingabe über File I/O
- Eingabe-File-Pfad als Argument übergeben
- Input-File darf nicht verändert werden
  - Bei berechtigtem Bedarf vorher Rücksprache halten
- Komplette Ausgabe über File I/O
- Jeglicher nicht-triviale Speicher auf Heap
- Speicher-Allokation passend wählen, nicht unnötig groß
- Ausreichend Kommentare schreiben

Was muss abgegeben werden?

Abgabe

- 2er Gruppen
- Compiler darf keine warnings ausgeben
  - Zu verwendende Flags: `-Wall -Wextra`
- Archiv
  - Quellcode
  - Keine(!) Objektdateien, ausführbaren Dateien oder ähnliches
  - Input-Files (Trivialbeispiel + Test-File)
  - Generiertes Output-File
  - Readme
    - Compile-Aufrufe
    - Terminal-Output von Programmausführung
    - Ausführung mit Valgrind

Bewertung

- Eigenarbeit, keine Copy-Paste-Lösungen oder KI-Generierung
- Pünktliche Abgabe bis 08.07.2026, 23:59 Uhr per Mail
  - jannek.squar@uni-hamburg.de
- Vorstellung am 15.07.2026, 14:00 - 16:00Uhr (s.t.!)
  - Max. 10 Minuten pro Gruppe
- Korrektheit
- Robustheit (z.B. bei falscher Eingabe)
- Memory-Management
- Error-Checking
- Präsentation (vorher absprechen, wer was vorstellt)
  - Aber: Jeder muss alles erklären können

Ablauf

- Idealerweise bereits jetzt Gruppenbildung
- Themenvorstellung
- Verteilung der Themen, FCFS
- Abgabefrist: 08.07.2026, 23:59 Uhr
- Bei Unklarheiten oder im Zweifel rechtzeitig vorher fragen!
- Fragen?

Table of Contents

Überfällige Vermögenssteuer

Beschreibung:
Es sollen die k reichsten Studierenden an einer Universität bestimmt werden. Der Input ist eine Inventarliste, in der die Besitztümer einzelner Studierender aufgelistet werden sowie eine Zahl k, die angibt, wie viele der reichsten Studierenden berücksichtigt werden sollen. Leerzeilen trennen die Besitztümer der einzelnen Studierenden.

Ausgabe:
Die Summe der Besitztümer der k reichsten Studierenden. Außerdem Ausgabe des Durchschnittsvermögens inklusive Standardabweichung aller Studierenden

Trivialbeispiel

```
1   1000
2   2000
3   foo
4   3000
5
6   4000
7   d
8
9   5000
10  abc
11  6000
12
13  7000
14  8000
15  9000
16
17  10000
```

- Erster Studi: 6000
- Zweiter Studi: 4000
- Dritter Studi: 11000
- Vierter Studi: 24000
- Fünfter Studi: 10000

Lösung: $24000 + 11000 + 10000 = 45000$ für $k = 3$,

Durchschnittsvermögen: 11000, Standardabweichung ≈ 6986

Trivialbeispiel

```
1   1000
2   2000
3   foo
4   3000
5
6   4000
7   d
8
9   5000
10  abc
11  6000
12
13  7000
14  8000
15  9000
16
17  10000
```

- Erster Studi: 6000
- Zweiter Studi: 4000
- Dritter Studi: 11000
- Vierter Studi: 24000
- Fünfter Studi: 10000

Lösung: $24000 + 11000 + 10000 = 45000$ für $k = 3$,

Durchschnittsvermögen: 11000, Standardabweichung ≈ 6986

Niemand hat die Absicht zu schummeln...

Beschreibung:
Auf einem Stein-Schere-Papier Turnier sind Spiele manipuliert worden, um Wetten zu gewinnen. Um keinen Verdacht zu erregen, soll aber nicht jedes Spiel gewonnen werden. Der Input gibt an, wie die Spiele ausgehen sollen. Die erste Spalte gibt den Zug des Gegners an, die zweite wie das Spiel für einen selbst ausgehen soll.

A Stein (1 Punkt)	X verlieren (0 Punkte)
B Papier (2 Punkte)	Y unentschieden (3 Punkte)
C Schere (3 Punkte)	Z gewinnen (6 Punkte)

Die Gesamtpunktzahl ergibt sich aus dem Spielausgang plus der Punkte für den eigenen(!) Zug

Ausgabe:
Gesamtpunktzahl mit Gewinn-Statistik

Trivialbeispiel

```
1 A Y
2 B X
3 C Z
```

- Erstes Spiel: Gegner spielt Stein, Ergebnis soll unentschieden (3 Punkte) sein -> Stein (1 Punkt)
- Zweites Spiel: Gegner spielt Papier, Ergebnis soll verloren (0 Punkte) sein -> Stein (1 Punkt)
- Drittes Spiel: Gegner spielt Schere, Ergebnis soll gewonnen (6 Punkte) sein -> Stein (1 Punkt)

Lösung: 4 + 1 + 7 = 12 Punkte

Trivialbeispiel

```
1 A Y
2 B X
3 C Z
```

- Erstes Spiel: Gegner spielt Stein, Ergebnis soll unentschieden (3 Punkte) sein -> Stein (1 Punkt)
- Zweites Spiel: Gegner spielt Papier, Ergebnis soll verloren (0 Punkte) sein -> Stein (1 Punkt)
- Drittes Spiel: Gegner spielt Schere, Ergebnis soll gewonnen (6 Punkte) sein -> Stein (1 Punkt)

Lösung: 4 + 1 + 7 = 12 Punkte

Studis auf Exkursion

Beschreibung:
Studierende fahren auf Exkursion und müssen ihre Rucksäcke packen. Jeweils drei Studierende bilden eine Gruppe, jede Person packt einen Rucksack. In jeder Gruppe gibt einen Gegenstand, der in allen drei Rucksäcken vorkommt. Der Buchstabe bestimmt dabei die Priorität des Gegenstands:

a-z 1-26 Punkte

A-Z 27-52 Punkte

Ausgabe:
Summe der Prioritäten der Gegenstände aller Gruppen, die jeweils in allen drei Rucksäcken einer Gruppe vorkommen, sowie Liste der gewählten Gegenstände.

Trivialbeispiel

```
1 vJrwpWtwJgWrhcsFMMfFFhFp
2 jqHRNqRjqzjGDLGLrsFMfFZSrLrFZsSL
3 PmmdzqPrVvPwwTWBwg
4 wMqvLMZHhHMvwLHjbvcjnnSBnvTQFn
5 ttgJtRGJQctTZtZT
6 CrZsJsPPZsGzwwsLwLmpwMDw
```

- Jede Zeile ist ein Rucksack
- Jeder Buchstabe ist ein Gegenstand
- Je drei Zeilen gehören zu einer Gruppe
- In der ersten Gruppe ist r der gemeinsame Gegenstand (18 Punkte)
- In der zweiten Gruppe ist Z der gemeinsame Gegenstand (52 Punkte)

Lösung: 18 (r) + 52 (Z) = 70 Punkte

Trivialbeispiel

```
1 vJrwpWtwJgWrhcsFMMfFFhFp
2 jqHRNqRjqzjGDLGLrsFMfFZSrLrFZsSL
3 PmmdzqPrVvPwwTWBwg
4 wMqvLMZHhHMvwLHjbvcjnnSBnvTQFn
5 ttgJtRGJQctTZtZT
6 CrZsJsPPZsGzwwsLwLmpwMDw
```

- Jede Zeile ist ein Rucksack
- Jeder Buchstabe ist ein Gegenstand
- Je drei Zeilen gehören zu einer Gruppe
- In der ersten Gruppe ist r der gemeinsame Gegenstand (18 Punkte)
- In der zweiten Gruppe ist Z der gemeinsame Gegenstand (52 Punkte)

Lösung: 18 (r) + 52 (Z) = 70 Punkte

Optimiertes Putzen

Beschreibung:
Nach der Exkursion müssen die Studierenden paarweise ihre Zimmer putzen. Die Zimmer sind in durchnummerierte Sektoren unterteilt. Jede Person bekommt einen Bereich (zusammenhängende Sektoren) zum Putzen zugewiesen, der durch zwei Zahlen repräsentiert wird: Anfangs- und Endsektor. Die Bereiche zweier Studierender können sich überlappen, müssen es aber nicht.

Ausgabe:
Summe wie oft generell Sektoren innerhalb der Studierendenpaare überlappen sowie Gesamtzahl überlappender Sektoren

Trivialbeispiel

```
2-4,6-8
2-3,4-5
5-7,7-9
2-8,3-7
6-6,4-6
2-6,4-8
```

- Paar 1: kein Überlapp
- Paar 2: kein Überlapp
- Paar 3: Überlapp Sektor 7
- Paar 4: Überlapp Sektoren 3-7
- Paar 5: Überlapp Sektor 6
- Paar 6: Überlapp Sektoren 4-6

Lösung: Überlapp bei 4 Paaren, insgesamt 1 + 5 + 1 + 3 = 10 überlappende Sektoren

Trivialbeispiel

```
2-4,6-8
2-3,4-5
5-7,7-9
2-8,3-7
6-6,4-6
2-6,4-8
```

- Paar 1: kein Überlapp
- Paar 2: kein Überlapp
- Paar 3: Überlapp Sektor 7
- Paar 4: Überlapp Sektoren 3-7
- Paar 5: Überlapp Sektor 6
- Paar 6: Überlapp Sektoren 4-6

Lösung: Überlapp bei 4 Paaren, insgesamt 1 + 5 + 1 + 3 = 10 überlappende Sektoren

Beschreibung:
Die Professorin der Arbeitsgruppe lässt ihre SHKs die Kisten mit Abschlussarbeiten umschichten. Dabei gibt sie konkrete Anweisungen vor, in welcher Abfolge die Kisten umgeräumt werden sollen. Wenn mehrere Kisten auf einmal umgeräumt werden, sollen sie ihre Reihenfolge beibehalten.

Ausgabe:
Darstellung des Endzustands

Trivialbeispiel

```
    [D]
[N] [C]
[Z] [M] [P]
 1   2   3

move 1 from 2 to 1
move 3 from 1 to 3
move 2 from 2 to 1
move 1 from 1 to 2
```

```
[D]
[N] [C]
[Z] [M] [P]
 1   2   3
```

```
      [D]
      [N]
  [C] [Z]
  [M] [P]
1   2   3
```

```
      [D]
      [N]
[C]   [Z]
[M]   [P]
 1   2   3
```

```
      [D]
      [N]
      [Z]
[M] [C] [P]
 1   2   3
```

Beschreibung:
In einem Datensatz sei eine Nachricht codiert. Der Beginn der Nachricht wird durch einen Marker markiert, die Nachricht selbst ist für diese Aufgabe aber irrelevant. Ein Marker besteht aus einer bestimmten Anzahl von aufeinanderfolgenden, unterschiedlichen Zeichen; jedes Zeichen darf innerhalb des Markers nur einmal vorkommen. Je mehr Zeichen ein Marker enthält, desto weniger Marker dieser Länge wird es im gegebenen Datensatz geben.

Ausgabe:
Länge, sodass es genau einen Marker dieser Länge im Datensatz gibt, sowie die End-Position des Markers im Datensatz.

Trivialbeispiel

```
mjqjspjmqmsmgsjslj
```

- Marker Länge 1: #18 (jedes Zeichen für sich)
- Marker Länge 2: #17 (keine zwei aufeinanderfolgenden Zeichen sind gleich)
- Marker Länge 3: #12 (jqj ist z.B. kein Marker)
- Marker Länge 4: #4 (Ende nach Zeichen 6, 8, 9 oder 15)
- Marker Länge 5: #1 (Ende nach Zeichen 9)
- Marker Länge 6: #0 (kein Marker dieser Länge)
- keine längeren Marker mehr möglich

Lösung: Marker Länge 5, Position 9

Trivialbeispiel

```
mjqjspjmqmsmgsjslj
```

- Marker Länge 1: #18 (jedes Zeichen für sich)
- Marker Länge 2: #17 (keine zwei aufeinanderfolgenden Zeichen sind gleich)
- Marker Länge 3: #12 (jqj ist z.B. kein Marker)
- Marker Länge 4: #4 (Ende nach Zeichen 6, 8, 9 oder 15)
- Marker Länge 5: #1 (Ende nach Zeichen 9)
- Marker Länge 6: #0 (kein Marker dieser Länge)
- keine längeren Marker mehr möglich

Lösung: Marker Länge 5, Position 9

Das Geomatikum zieht um

Beschreibung:
Eine Stadt besteht aus unterschiedlich hohen Gebäuden auf einem regelmäßigen Raster. Da das Haus der Erde niemals fertig wird, soll das Geomatikum nun in ein anderes Gebäude umziehen. Dabei soll das Gebäude mit der besten Aussicht ausgewählt werden. Die Sichtweiten-Metrik eines bestimmten Gebäudes bestimmt sich aus dem Produkt der sichtbaren Gebäude in alle vier Himmelsrichtungen (vom ausgewählten Gebäude aus gesehen). Ursprung des Koordinaten-Systems liegt „oben links“.

Ausgabe:
Koordinaten des Gebäudes mit der besten Sichtweiten-Metrik sowie die Sichtweiten-Metrik dieses Gebäudes.

Trivialbeispiel I

```
30373
2551 2
65332
33549
35390
```

Beispiel Gebäude auf Position (1,2) mit Höhe 5:

- Links: 1 sichtbares Gebäude (Höhe 5)
- Rechts: 2 sichtbare Gebäude (Höhen 1 und 2)
- Oben: 1 sichtbares Gebäude (Höhe 3)
- Unten: 2 sichtbare Gebäude (Höhen 3 und 5)
- Sichtweiten-Metrik: $1 * 2 * 1 * 2 = 4$

Trivialbeispiel II

```
30373
25512
65332
33549
35390
```

Beispiel Gebäude auf Position (3,2) mit Höhe 5:

- Links: 2 sichtbare Gebäude (Höhen 3 und 3)
- Rechts: 2 sichtbare Gebäude (Höhen 4 und 9)
- Oben: 2 sichtbare Gebäude (Höhen 3 und 5)
- Unten: 1 sichtbares Gebäude (Höhe 3)
- Sichtweiten-Metrik: $2 * 2 * 2 * 1 = 8$

Lösung: Sichtweiten-Metrik 8, Position (3,2)

Trivialbeispiel II

```
30373
25512
65332
33549
35390
```

Beispiel Gebäude auf Position (3,2) mit Höhe 5:

- Links: 2 sichtbare Gebäude (Höhen 3 und 3)
- Rechts: 2 sichtbare Gebäude (Höhen 4 und 9)
- Oben: 2 sichtbare Gebäude (Höhen 3 und 5)
- Unten: 1 sichtbares Gebäude (Höhe 3)
- Sichtweiten-Metrik: $2 * 2 * 2 * 1 = 8$

Lösung: Sichtweiten-Metrik 8, Position (3,2)

Stadtführung

Beschreibung:
Neun Studis auf Exkursion in einer fremden Stadt wollen eine Stadtführung machen. Damit niemand verloren geht, geht der Tour-Guide voran und die anderen folgen, wobei sich alle an den Händen fassen und so eine lange Schlange bilden.

Ausgabe:
Wie viele verschiedene Positionen hat die letzte studierende Person in der Schlange mindestens einmal besucht?

Beispiele für erlaubte Positionen

Menschen, die sich an den Händen halten, müssen immer benachbart zueinander stehen, d.h. es darf keine Lücke zwischen ihnen geben. Diagonale Nachbarschaft reicht dabei aus, sie dürfen außerdem auch auf der gleichen Position stehen.

```
....
.1T.
....
```

```
....
.1..
.T.
....
```

```
....
.T..
..1.
....
```

```
...
.T. (T bedeckt Studi 1)
...
```

Gummiband-Regeln

Wenn sich Lücken bilden, muss die hintere studierende Person nachziehen, bis die Schlange wieder geschlossen ist.

```
.....    .....    .....
.1T.. -> .1.T. -> ..1T.
.....    .....    .....
```

```
...    ...    ...
.1.    .1.    ...
.T. -> ... -> .1.
...    .T.    .T.
...    ...    ...
```

```
.....    .....    .....
.....    ..T..    ..T..
..T.. -> ..... -> ..1..
.1...    .1...    .....
.....    .....    .....
```

```
.....    .....    .....
.....    .....    .....
..T.. -> ...T. -> ..1T.
.1...    .1...    .....
.....    .....    .....
```

Trivialbeispiel I

```
R 5
U 8
L 8
D 3
R 17
D 10
L 25
U 20
```

```
== Initial State ==

..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........T...............
..........................
..........................
..........................
..........................
..........................
(T überdeckt 1, 2, 3, 4, 5, 6, 7, 8, 9, s)
```

```
== R 5 ==

..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........54321T..........
..........................
..........................
..........................
..........................
..........................
(5 überdeckt 6, 7, 8, 9, s)
```

Trivialbeispiel II

```
== U 8 ==

..........................
..........................
..........................
..........................
..........................
..........................
..........................
.................T........
.................1........
................2.........
................3.........
...............54.........
..............6...........
.............7............
............8.............
...........9..............
..........................
..........................
..........................
..........................
..........................
(9 überdeckt s)
```

```
== L 8 ==

..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
........T1234.............
............5.............
............6.............
............7.............
............8.............
............9.............
..........................
..........................
..........s...............
..........................
..........................
..........................
..........................
..........................
```

```
== D 3 ==

..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........................
..........2345............
........1...6.............
........T...7.............
............8.............
............9.............
..........................
..........................
...........s..............
..........................
..........................
..........................
..........................
..........................
```

Trivialbeispiel III

```
== R 17 ==

.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
................987654321T
.........................
.........................
.........................
.........................
.........................
..........s..............
.........................
.........................
.........................
.........................
.........................
```

```
== D 10 ==

.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........s.........98765
.......................4
.......................3
.......................2
.......................1
.......................T
```

```
== L 25 ==

.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
.........................
..........s..............
.........................
.........................
.........................
.........................
T123456789...............
```

Trivialbeispiel IV

```
== U 20 ==

T.......................
1.......................
2.......................
3.......................
4.......................
5.......................
6.......................
7.......................
8.......................
9.......................
.......................
.......................
.......................
.......................
.......................
...........s...........
.......................
.......................
.......................
.......................
.......................
```

```
.......................
.......................
.......................
.......................
.......................
.......................
.......................
.......................
.......................
#......................
#.............###......
#............#...#.....
.#.........#.....#......
..#.......#.....#.......
...#.....#.....#........
....#.....s.....#.......
.....#...........#......
......#.........#.......
.......#.......#........
........#.....#.........
.........#####..........
```

Lösung: Studi 9 hat 36 Positionen besucht

Beschreibung:
Mit einem einfachen Befehlssatz soll ein Pixelbild gezeichnet werden. Werte in einem Register bestimmen die horizontale Position eines 3 Pixel breiten Blocks, der über einen 40 Pixel breiten CRT-Bildschirm läuft. Befehle benötigen zur Ausführung eine bestimmte Anzahl von Zyklen, während derer das Register nicht verändert wird. Die Bildschirmausgabe erfolgt parallel zur Programmausführung.

Ausgabe:
Das finale Bild, das durch die Befehle gezeichnet wird.

Bildschirm

Die Programmausführung ist in Zyklen unterteilt, in jedem Zyklus wird das nächste Pixel gezeichnet:

- #, wenn das zu zeichnende Pixel von dem 3 Pixel breiten Block bedeckt ist
- ., wenn das zu zeichnende Pixel nicht von dem Block bedeckt ist

```
Zyklus   1 -> ######################################## <- Zyklus  40
Zyklus  41 -> ######################################## <- Zyklus  80
Zyklus  81 -> ######################################## <- Zyklus 120
Zyklus 121 -> ######################################## <- Zyklus 160
Zyklus 161 -> ######################################## <- Zyklus 200
Zyklus 201 -> ######################################## <- Zyklus 240
```

Der verwendete Befehlssatz besteht aus zwei Befehlen:

- noop: dauert 1 Zyklus, verändert das Register nicht
- addx V: dauert 2 Zyklen, erhöht am Ende des zweiten Zyklus das Register um den Wert V (V kann negativ sein)

Beispielprogramm:

```
noop
addx 3
addx -5
noop
```

Zyklus	Aktuelle Operation	Register
1	noop	1
2	addx 3	1
3	addx 3	1
4	addx -5	4
5	addx -5	4
6	noop	-1

Beispielprogramm I

```
addx 15
addx -11
addx 6
addx -3
```

```
Block Position: ###.................................

Beginn Zyklus   1: Beginne addx 15
Während Zyklus  1: CRT setzt Pixel an Position 0
Bildschirm:     #

Während Zyklus  2: CRT setzt Pixel an Position 1
Bildschirm:     ##
Nach Zyklus     2: Vollende addx 15 (Register X ist 16)
Block Position: ..............###.....................

Beginn Zyklus   3: Beginne addx -11
Während Zyklus  3: CRT setzt Pixel an Position 2
Bildschirm:     ##.

Während Zyklus  4: CRT setzt Pixel an Position 3
Bildschirm:     ##..
Nach Zyklus     4: Vollende addx -11 (Register X ist 5)
Block Position: ....###.............................

Beginn Zyklus   5: Beginne addx 6
Während Zyklus  5: CRT setzt Pixel an Position 4
Bildschirm:     ##..#
```

Beispielprogramm II

```
addx 15
addx -11
addx 6
addx -3
```

```
Während Zyklus  6: CRT setzt Pixel an Position 5
Bildschirm:     ##..##
Nach Zyklus     6: Vollende addx 6 (Register X ist 11)
Block Position: ..........###..........................

Beginn Zyklus   7: Beginne addx -3
Während Zyklus  7: CRT setzt Pixel an Position 6
Bildschirm:     ##..##.

Während Zyklus  8: CRT setzt Pixel an Position 7
Bildschirm:     ##..##..
Nach Zyklus     8: Vollende addx -3 (Register X ist 8)
Block Position: .......###............................
```

Defekter Fahrstuhl

Beschreibung:
Der Fahrstuhl des Gebäudes ist leider defekt, dennoch müssen die Studis zum Seminarraum im obersten Stockwerk gelangen. Zum Glück gibt es unzählige Treppen zwischen einzelnen Stockwerken. Die Buchstaben geben die Höhe der Stockwerke an, wobei a die niedrigste und z die höchste Höhe darstellt. Die Studis können nur von einem Stockwerk zu einem anderen wechseln, wenn die Höhe des Zielstockwerks maximal um 1 höher ist als die Höhe des aktuellen Stockwerks. D.h. $m \rightarrow n$ ist erlaubt, $m \rightarrow o$ ist nicht erlaubt. Startpunkt S liegt auf Höhe a, Zielpunkt E liegt auf Höhe z.

Ausgabe:
Anzahl Schritte des kürzesten Weges vom Startpunkt S zum Zielpunkt E und grafische Darstellung des Pfades.

Trivialbeispiel

```
Sabqponm
abcryxxl
accszExk
acctuvwj
abdefghi
```

```
v..v<<<<
>v.vv<<^
.>vv>E^^
..v>>>^^
..>>>>>^
```

- $<$ Bewegung nach links
- $>$ Bewegung nach rechts
- $\wedge$ Bewegung nach oben
- $\vee$ Bewegung nach unten
- . nicht benutzte Position

Lösung: 31 Schritte

Trivialbeispiel

```
Sabqponm
abcryxxl
accszExk
acctuvwj
abdefghi
```

```
v..v<<<<
>v.vv<<^
.>vv>E^^
..v>>>^^
..>>>>>^
```

- $<$ Bewegung nach links
- $>$ Bewegung nach rechts
- $\wedge$ Bewegung nach oben
- $\vee$ Bewegung nach unten
- . nicht benutzte Position

Lösung: 31 Schritte

Flaschenpfand im CT

Beschreibung:
Die Pfandflaschen im CT nehmen überhand und müssen entfernt werden. Die Flaschen sind in einem Raster angeordnet. Eine Flasche kann nur dann gefahrlos entfernt werden, wenn auf den umliegenden acht Feldern maximal drei weitere Flaschen stehen. Das Entfernen einer Flasche kann dazu führen, dass weitere Flaschen gefahrlos entfernt werden können.

Ausgabe:
Die Anzahl an Flaschen, die iterativ gefahrlos entfernt werden können sowie Darstellung des Endzustands.

Trivialbeispiel

Ausgangslage

```
..@@.@@@@.
@@@.@.@.@@
@@@@@.@.@@
@.@@@@..@.
@@.@@@@.@@
.@@@@@@@.@
.@.@.@.@@@
@.@@@.@@@@
.@@@@@@@@.
@.@.@@@.@.
```

1. Iteration

```
..xx.xx@x.
x@@.@.@.@@
@@@@@.x.@@
@.@@@@..@.
x@.@@@@.@x
.@@@@@@@.@
.@.@.@.@@@
x.@@@.@@@@
.@@@@@@@@.
x.x.@@@.x.
```

2. Iteration

```
.......x..
.@@.x.x.@x
x@@@@...@@
x.@@@@..x.
.@.@@@@.x.
.x@@@@@@.x
.x.@.@.@@@
..@@@.@@@@
.x@@@@@@@.
....@@@...
```

9. Iteration

```
..........
..........
..........
...x@@....
...@@@@...
...@@@@@..
...@.@.@@.
...@@.@@@.
...@@@@@..
....@@@...
```

Lösung: 43 Flaschen wurden entfernt

Trivialbeispiel

Ausgangslage

```
..@@.@@@@.
@@@.@.@.@@
@@@@@.@.@@
@.@@@@..@.
@@.@@@@.@@
.@@@@@@@.@
.@.@.@.@@@
@.@@@.@@@@
.@@@@@@@@.
@.@.@@@.@.
```

1. Iteration

```
..xx.xx@x.
x@@.@.@.@@
@@@@@.x.@@
@.@@@@..@.
x@.@@@@.@x
.@@@@@@@.@
.@.@.@.@@@
x.@@@.@@@@
.@@@@@@@@.
x.x.@@@.x.
```

2. Iteration

```
.......x..
.@@.x.x.@x
x@@@@...@@
x.@@@@..x.
.@.@@@@.x.
.x@@@@@@.x
.x.@.@.@@@
..@@@.@@@@
.x@@@@@@@.
....@@@...
```

9. Iteration

```
..........
..........
..........
...x@@....
...@@@@...
...@@@@@..
...@.@.@@.
...@@.@@@.
...@@@@@..
....@@@...
```

Lösung: 43 Flaschen wurden entfernt

Trivialbeispiel

Ausgangslage

```
..@@.@@@@.
@@@.@.@.@@
@@@@@.@.@@
@.@@@@..@.
@@.@@@@.@@
.@@@@@@@.@
.@.@.@.@@@
@.@@@.@@@@
.@@@@@@@@.
@.@.@@@.@.
```

1. Iteration

```
..xx.xx@x.
x@@.@.@.@@
@@@@@.x.@@
@.@@@@..@.
x@.@@@@.@x
.@@@@@@@.@
.@.@.@.@@@
x.@@@.@@@@
.@@@@@@@@.
x.x.@@@.x.
```

2. Iteration

```
.......x..
.@@.x.x.@x
x@@@@...@@
x.@@@@..x.
.@.@@@@.x.
.x@@@@@@.x
.x.@.@.@@@
..@@@.@@@@
.x@@@@@@@.
....@@@...
```

9. Iteration

```
..........
..........
..........
...x@@....
...@@@@...
...@@@@@..
...@.@.@@.
...@@.@@@.
...@@@@@..
....@@@...
```

Lösung: 43 Flaschen wurden entfernt

Trivialbeispiel

Ausgangslage

```
..@@.@@@@.
@@@.@.@.@@
@@@@@.@.@@
@.@@@@..@.
@@.@@@@.@@
.@@@@@@@.@
.@.@.@.@@@
@.@@@.@@@@
.@@@@@@@@.
@.@.@@@.@.
```

1. Iteration

```
..xx.xx@x.
x@@.@.@.@@
@@@@@.x.@@
@.@@@@..@.
x@.@@@@.@x
.@@@@@@@.@
.@.@.@.@@@
x.@@@.@@@@
.@@@@@@@@.
x.x.@@@.x.
```

2. Iteration

```
.......x..
.@@.x.x.@x
x@@@@...@@
x.@@@@..x.
.@.@@@@.x.
.x@@@@@@.x
.x.@.@.@@@
..@@@.@@@@
.x@@@@@@@.
....@@@...
```

9. Iteration

```
..........
..........
..........
...x@@....
...@@@@...
...@@@@@..
...@.@.@@.
...@@.@@@.
...@@@@@..
....@@@...
```

Lösung: 43 Flaschen wurden entfernt

Trivialbeispiel

Ausgangslage

```
..@@.@@@@.
@@@.@.@.@@
@@@@@.@.@@
@.@@@@..@.
@@.@@@@.@@
.@@@@@@@.@
.@.@.@.@@@
@.@@@.@@@@
.@@@@@@@@.
@.@.@@@.@.
```

1. Iteration

```
..xx.xx@x.
x@@.@.@.@@
@@@@@.x.@@
@.@@@@..@.
x@.@@@@.@x
.@@@@@@@.@
.@.@.@.@@@
x.@@@.@@@@
.@@@@@@@@.
x.x.@@@.x.
```

2. Iteration

```
.......x..
.@@.x.x.@x
x@@@@...@@
x.@@@@..x.
.@.@@@@.x.
.x@@@@@@.x
.x.@.@.@@@
..@@@.@@@@
.x@@@@@@@.
....@@@...
```

9. Iteration

```
..........
..........
..........
...x@@....
...@@@@...
...@@@@@..
...@.@.@@.
...@@.@@@.
...@@@@@..
....@@@...
```

Lösung: 43 Flaschen wurden entfernt

Rookie Numbers

Beschreibung:
Die Verwaltung stellt die Organisation der Studi-Akten auf ein neues System um, bei dem nummerierte Ordner angelegt werden. Hierfür werden aus ominösen Gründen in jeder Zeile zwölf Ziffern ausgewählt, die dann zu einer Zahl zusammengesetzt werden. Die Auswahl der Ziffern erfolgt dabei so, dass die größtmögliche Zahl entsteht. Die Ziffern müssen dabei in der Reihenfolge bleiben, in der sie in der Zeile vorkommen, es dürfen also keine Ziffern vertauscht werden.

Ausgabe:
Summe der größten Zahlen, die in jeder Zeile des gegebenen Datensatzes gefunden werden können, sowie die gefundenen Zahlen selbst.

Trivialbeispiel

```
987654321111111
811111111111119
234234234234278
818181911112111
```

Größten Zahlen

- Zeile 1: 987654321111
- Zeile 2: 811111111119
- Zeile 3: 434234234278
- Zeile 4: 888911112111

Lösung: $987654321111 + 811111111119 + 434234234278 + 888911112111 = 3121910778619$

Trivialbeispiel

```
987654321111111
811111111111119
234234234234278
818181911112111
```

Größten Zahlen

- Zeile 1: 987654321111
- Zeile 2: 811111111119
- Zeile 3: 434234234278
- Zeile 4: 888911112111

Lösung: $987654321111 + 811111111119 + 434234234278 + 888911112111 = 3121910778619$

Table of Contents

Zuteilung der Projekte

- 1: Vermögenssteuer
- 2: Schere-Stein-Papier
- 3: Rucksack-Organisation
- 4: Überlappendes Putzen
- 5: Studis von Hanoi
- 6: Nachrichtenmarker
- 7: Neues Geomatikum
- 8: Stadtführung
- 9: Pixelkunst
- 10: Treppenauswahl
- 11: Flaschenpfand
- 12: Größte Nummer