

C-ÜBUNGSAUFGABEN FELDER 1

Bemerkungen:

Die Feldlänge soll vom Programmierer ohne aufwändige Bemühungen an genau einer Stelle leicht zu verändern sein (und muß dann neu compiliert werden).

1) Wärmeverbrauch

1.1)

a) Es soll der monatliche Wärmeverbrauch (in GJ) eines Personenhaushalts eingegeben und in einem Feld der Länge 12 abgespeichert werden. Es soll außerdem verhindert werden, dass der Anwender Daten für mehr als 12 Monate eingeben kann.

Geben Sie die eingegebenen Daten zur Kontrolle nochmals auf dem Bildschirm aus.

b) Berechnen Sie den mittleren Wärmeverbrauch (pro Jahr) und geben Sie das Ergebnis auf dem Bildschirm aus.

c) Berechnen Sie den maximalen Wärmeverbrauch (und den zugehörigen Monat) und geben Sie das Ergebnis auf dem Bildschirm aus.

d) Berechnen Sie den minimalen Wärmeverbrauch (und den zugehörigen Monat) und geben Sie das Ergebnis auf dem Bildschirm aus.

e) Ein GJ Wärmeenergie kostet bei den Stadtwerken Nürtingen 14,27 Euro.

Berechnen Sie (mit Hilfe der Daten bei a)) die monatlichen Wärmeverbrauchskosten und speichern diese in einem neuen Feld ab. Geben Sie dann den Inhalt dieses Feldes auf dem Bildschirm aus.

1.2)

Um Daten (wie z.B. bei der vorigen Aufgabe) besser zu schützen, sollen sie verschlüsselt werden.

a) Der Anwender soll jeden Monat durch ein eindeutiges Zeichen (z.B. Januar durch r) kennzeichnen, d.h. er gibt diese 12 Zeichen über Tastatur ein und speichert sie in einem Feld ab. (Wie groß wählen Sie die Feldlänge ?)

Geben Sie die eingegebenen Daten (z.B. als Zeichenkette) zur Kontrolle nochmals auf dem Bildschirm aus.

b) Der Anwender soll nachprüfen können, ob ein bestimmtes Zeichen (das er über Tastatur eingibt) in dieser Zeichenkette vorkommt.

c) Aus sicherheitstechnischen Gründen sollen die Daten bei a) nochmals in umgekehrter Reihenfolge in ein neues Feld kopiert werden und der Inhalt des neuen Feldes (mit dem 0. Element beginnend) auf dem Bildschirm ausgegeben werden.

Beispiel:

monatslisteAlt:

r	f	m	a	i	j	u	g	s	o	n	d
---	---	---	---	---	---	---	---	---	---	---	---

monatslisteNeu:

d	n	o	s	g	u	j	i	a	m	f	r
---	---	---	---	---	---	---	---	---	---	---	---

2)

In einem Integer-Feld soll der kleinste Abstand (ein Abstand ist immer positiv) zweier benachbarter Zahlen und der Index der ersten Zahl (vom ersten Vorkommen, falls es mehrere benachbarte Zahlen mit gleichem Abstand gibt) berechnet werden.

3)

Von einer Zahlenfolge soll ermittelt werden, ob sie fallend, streng fallend (d.h. fallend, aber keine 2 gleichen Zahlen hintereinander vorkommend) wachsend, streng wachsend (d.h. wachsend, aber keine 2 gleichen Zahlen hintereinander vorkommend) oder unsortiert ist.

4)

Die über Tastatur eingegebenen Zeichen sollen in einem Feld der Länge LEN (z.B. LEN = 5) abgespeichert werden. Werden mehr als LEN-1 Zeichen eingegeben, so werden nur LEN-1 abgespeichert und mit '\0' abgeschlossen.

Werden weniger als LEN-1 Zeichen eingegeben, so werden diese abgespeichert und mit '\0' abgeschlossen.

Wird dies durch das folgende Programm gemacht ?

Wenn nicht, dann zeigen Sie dies an einem konkreten Beispiel. Schreibtischtest!

```
#include "stdafx.h"
#include "stdio.h"

int main(int argc, char* argv[]){
    const int len=3;
    char namen[len];
    int i;

    printf("Bitte Namen eingeben\n");
    for(i=0; i<len-1; i++){
        scanf("%c",&namen[i]);
        if(namen[i]==10){
            break;
        }
    }
    fflush(stdin);
    namen[len-1]='\0';

    printf("Name=\n");
    printf("%s",namen);

    return 0;
}
```

5)

In einem eindimensionalen Zahlenfeld der Länge len soll von den ersten $n \leq \text{len}$ Zahlen festgestellt werden, ob diese alle verschieden sind.

Schreiben Sie ein dazugehöriges Programm.

6)

In einem eindimensionalen Zahlenfeld der Länge len soll das Maximum (die größte Zahl) ermittelt werden.

Außerdem soll auch der Index der maximalen Zahl berechnet werden.

Falls es mehrere gleich große maximale Zahlen gibt, soll ein Feld mit deren Indizes berechnet werden. Schreiben Sie ein dazugehöriges Programm.

Beispiele:

[4, -3, 4, 4] --> max = 4, Indizes = [0, 2, 3]

[2, 7, 5, -1] --> max = 7, Indizes = [1]

7) Random Walk

0	0	0	0	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

a) Es sollen die Bewegungen eines Betrunkenen simuliert (wissenschaftlich: random walk), werden, der aus einer Kneipe kommt, sich mit Hilfe des Zufalls fortbewegt und hofft dadurch irgendwann daheim anzukommen:

Mit jeweils der Wahrscheinlichkeit 0,5 bewegt er sich in einem Array (von der Mitte ausgehend) um eine Zelle nach links bzw. um eine Zelle nach rechts. Nach jeder Bewegung wird in die jeweilige Zelle geschrieben, wie oft sie schon insgesamt besucht wurde.

Zu Beginn der Simulation sieht die Belegung des Array also - wie oben dargestellt - aus.

Wenn der Betrunkene an einem der beiden Ränder des Arrays angekommen ist, wird die Simulation beendet (d.h. bei Index = 0 oder Index = Arraylänge-1).

Um die Simulation anschauen zu können, soll nach jeder Bewegung der gesamte Inhalt des Arrays auf dem Bildschirm (einschließlich des Index der aktuellen Zelle, d.h. die Zellennummer, wo sich der Betrunkene gerade befindet) ausgegeben werden.

In einer anderen Version muß nach jedem Tastendruck das Programm alle Elemente des Arrays (einschließlich des Index der aktuellen Zelle) auf dem Bildschirm ausgegeben werden.

b) Machen Sie dieselbe Aufgabe für den Random Walk in einem zweidimensionalen Feld:

Mit jeweils der Wahrscheinlichkeit 0,25 bewegt sich der Betrunkene in dem zweidimensionalen Feld (von der Mitte ausgehend) um eine Zelle nach links, bzw. nach rechts, nach oben oder nach unten.

8) Simulation

Chamäleons können ihre Farbe wechseln. Sie tun dies allerdings nicht zur besseren Tarnung, sondern um mit Artgenossen zu kommunizieren.

7 Chamäleons leben gemeinsam in einem großen Terrarium. Vier von ihnen sind rot gefärbt, zwei blau und eins ist grün. Die Tiere können jede dieser drei Farben annehmen, aber keine weitere.

Begegnen sich zwei Tiere, kommt es bei beiden zu einem Farbwechsel – allerdings nur, wenn beide unterschiedlich gefärbt sind. Beide nehmen dann die dritte Farbe an.

Ein Beispiel: Treffen sich ein rotes und ein grünes Tier, sind sie danach beide blau.

Ist es möglich, dass nach einer gewissen Zeit alle Chamäleons im Terrarium dieselbe Farbe haben?

Doch wie ist die Situation, wenn im Terrarium sechs rote, zwei blaue und ein grünes Tier leben?

Simulieren Sie das Terrarium durch ein Feld und die zufälligen Treffen zweier Chamäleons durch den Zufall.

9)

Das Feld zahlenFeld[...] der Länge 10 besteht aus ganzen Zahlen. Von diesen soll das Maximum und das Minimum bestimmt werden. Diskutieren Sie den folgenden

Lösungsvorschlag:

```
...
max = zahlenFeld[0];
min = zahlenFeld[0];
for(i=0;i<10;i++){
    if(zahlenFeld[i] > max){
        max = zahlenFeld[i];
    }
    else{
        if(zahlenFeld[i] < min){
            min = zahlenFeld[i];
        }
    }
}
...
```

10) schwierig

Entwickeln Sie ein Programm, das einen Kilometerzähler (mit n=4 Stellen, zwischen denen sich jeweils das Zeichen # befindet) simuliert.

Der Kilometerzähler soll hochgezählt werden und der aktuelle Kilometerstand auf dem Bildschirm ausgegeben werden.

Benutzen Sie diese "Technik" um das Zahlenrätsel SEND+MORE=MONEY zu lösen.

Hinweis: (Verallgemeinerung)

n soll veränderlich sein (z.B. kann der Anwender den Wert von n über Tastatur eingeben).

Schreiben Sie das Programm so, daß die Anzahl der Schleifen nicht von n abhängig sein darf.

Benutzen Sie diese "Technik" um das Zahlenrätsel SEND+MORE=MONEY zu lösen.

11) Wo steckt der Fehler?

feld2 wird an feld1 angefügt. Das Ergebnis soll in feld stehen. Warum gibt es Probleme während der Laufzeit? (Compiler meckert aber nicht).

```
int main(){
    const int LEN = 20;
    char feld1[LEN]="V3";
    char feld2[LEN]="Gut";
    char feld [2*LEN];
    int i, m = 0, h=0;
    for ( i = 0; i < 2*LEN; i++)
    {
        if (feld1[m]!='\0'){
            feld[i] = feld1[m];
            m++;
        }
        else{
            feld[i] = feld2[h];
            h++;
        }
    }
    return(0);
}
```

12) Feldlänge bestimmen

Die Länge eines Feldes berechnen.

a)

Beispiel: LEN = 17

gegeben: (wird über Tastatur eingegeben)

Feld string:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	------	---	---	---	---	---	---	---

danach:

Länge des Feldes: 9

13) Entferne Zeichen an einer Position

Entfernt ein Zeichen an einer vorgegebenen Position.

a)

Beispiel:

gegeben: (wird über Tastatur eingegeben)

pos = 5

Feld string:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	----------	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld string:

v	3	s	e	h	g	u	t	'\0'	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	------	---	---	---	---	---	---	---	---

b) wie a) außer dass Ergebnis in neues Feld kommt.

Beispiel:

gegeben: (wird über Tastatur eingegeben)

pos = 5

Feld stringIn:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	----------	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld stringOut:

v	3	s	e	h	g	u	t	'\0'	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	------	---	---	---	---	---	---	---	---

14) Feld anfügen

Eine Zeichenfolge wird an die andere angefügt (engl: concatenate).

a)

Beispiel:

gegeben: (wird über Tastatur eingegeben)

Feld string1

v	d	r	e	i	'\0'	?	?	?	'?	?	?	?	?	?	?	?
---	---	---	---	---	------	---	---	---	----	---	---	---	---	---	---	---

Feld string2

g	u	t	'\0'	?	?	?
----------	----------	----------	------	---	---	---

danach

Feld string1

v	d	r	e	i	g	u	t	'\0'	'?	?	?	?	?	?	?	?
---	---	---	---	---	----------	----------	----------	------	----	---	---	---	---	---	---	---

b)

Die Aneinanderfügung von string1 und string2 (string1 + string2) werden nicht in string1, sondern in einem neuen Feld string3 gespeichert.

15) Zeichen suchen

- a) Berechnet den Index (das erstmalige Vorkommen) eines Zeichens in einer Zeichenfolge.
- b) Berechnet den Index (das letzte Vorkommen) eines Zeichens in einer Zeichenfolge.

16) Feld umgekehrt anordnen

Ein Feld wird in umgekehrter Reihenfolge angeordnet.

a)

Beispiel:

Feld string:

v	3	c	h	e	f	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld string:

f	e	h	c	3	v	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	---	------	---	---	---	---	---	---	---

17) Zeichen anfügen

Ein Zeichen wird rechts einer bestimmten Position eingefügt.

a)

Beispiel:

gegeben: (wird über Tastatur eingegeben)

pos = 1

zeichen = s

Feld string:

v	3	e	h	r	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld string:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	------	---	---	---	---	---	---

b) wie a) außer dass Ergebnis in neues Feld kommt.

Beispiel:

gegeben: (wird über Tastatur eingegeben)

pos = 1

zeichen = s

Feld stringIn:

v	3	e	h	r	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld stringOut:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	------	---	---	---	---	---	---

c) Ein Zeichen wird links einer bestimmten Position angefügt.

d) Ein bestimmtes (über Tastatur eingegeben) Zeichen wird in einem Feld gesucht (das erstmalige Vorkommen).

Links bzw. rechts (über Tastatur eingegeben) dieses gefundenen Zeichens wird ein bestimmtes Zeichen (über Tastatur eingegeben) eingefügt.

18) Zeichenfolge (Feld) anfügen

a) Eine Zeichenfolge wird rechts einer bestimmten Position eingefügt.

Wie vorige Aufgabe, außer dass statt eines Zeichens eine Zeichenfolge (Feld) rechts bzw. links einer Position eingefügt wird.

Beispiel:

gegeben: (wird über Tastatur eingegeben)

pos = 1

stringInsert = "sehr"

Feld string:

v	3	g	u	t	'\0'	?	?	?	?	?	?	?
---	---	---	---	---	------	---	---	---	---	---	---	---

danach:

Feld string:

v	3	s	e	h	r	g	u	t	'\0'	?	?	?
---	---	---	---	---	---	---	---	---	------	---	---	---

b) Eine Zeichenfolge wird links einer bestimmten Position eingefügt.

19) Felder mit Zahlen

Gegeben ist das **nicht** notwendig vollständig mit Zahlen gefüllte Feld der Feldlänge LEN (Konstante)

int numbers [LEN]

Da man irgendwo speichern muß, wie viele Zahlen sich in dem Feld befinden (es muß ja nicht vollständig mit Zahlen gefüllt worden sein), muß diese Anzahl irgendwo gespeichert werden.

Hier wird die Anzahl im Element 0 des Feldes gespeichert.

Beispiel: Es befinden sich aktuell 5 Zahlen in dem Feld numbers.

5	7	-1	3	-9	4											
---	---	----	---	----	---	--	--	--	--	--	--	--	--	--	--	--

a) Geben Sie genau 2 Anweisungen in C an, die an das Feldende (d.h. direkt nach der Zahl 4) die Zahl 2 anfügt (auf obiges Beispiel bezogen).

b) Geben Sie genau eine einzige Anweisung in C an, die die letzte Zahl "löscht" (auf obiges Beispiel bezogen).

c) Geben Sie genau eine einzige Anweisung in C an, die alle Zahlen "löscht". (auf obiges Beispiel bezogen).

d) Lösen Sie alle vorigen Aufgaben (angefangen bei "entferne Zeichen an einer Position"), aber nicht mit Zeichenfeldern, sondern mit Zahlenfeldern.

20) Schiffe versenken

Programmieren Sie das Spiel "Schiffe versenken".

Wer es nicht kennt: siehe Wikipedia

21) Skatkarten simulieren

Skat wird mit einem Kartenspiel aus 32 Karten gespielt.

Die Skatkarten bestehen aus 32 Einzelkarten und sind dabei in vier Farben zu je acht Karten unterteilt. Jede Farbe enthält acht Karten, die jeweils einen Wert haben.

Sieben (7), Acht (8), Neun (9), Zehn (10), Bube (2), Dame (3), König (4) und Ass (11).

Die 32 Skatkarten liegen in Form eines Stapels auf dem Tisch.

Für ein Spiel wie z.B. Black Jack ("17 + 4") werden Karten von diesem Stapel genommen.

Simulieren Sie diesen Kartenstapel durch ein Feld.

22) Zaubertricks

22.1)

Ein Würfel wird $n > 25$ Mal (z.B. $n=26$) geworfen und jeweils die Zahl nacheinander in einem Feld notiert. Auf einem der ersten Elemente des Feldes wird gestartet.

Von dort ausgehend wird um so viele Feldelemente weitergegangen, wie die Zahl auf dem Feldelement angibt. Das wird immer wieder gemacht, bis es nicht mehr weiter geht, d.h. bis man beim nächsten Zug über das Feldende hinaus kommen würde.

Ein Zauberer behauptet:

Egal, bei welcher der ersten Feldelemente man begonnen hat, am Ende gelangt man immer auf dem gleichen letzten Feldelement.

Beispiel: $n=26$

4	5	2	5	5	1	5	4	6	5	5	1	5	5	3	6	5	2	5	3	1	3	6	5	2	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Start bei $i=0$ (Wurf 4)

4 \rightarrow 5 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 6

Start bei $i=1$ (Wurf 5)

5 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 6

Start bei $i=2$ (Wurf 2)

2 \rightarrow 5 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 6

Simulieren Sie diesen Zaubertrick und prüfen nach, mit welcher Wahrscheinlichkeit der Zauberer recht hat.

22.2)

Es wird eine Münze mit 2 Seiten benutzt, auf denen S (wie schwarz) und W (wie weiß) steht. Programmtechnisch kann man auch 0 bzw. 1 wählen.

Der Zuschauer muß ein Dreiermuster wählen (z.B. WSW). Dann muß der Zauberer ein anderes Dreiermuster auswählen (z.B. WWS).

Dann wird die obige Münze "hineichend" oft geworfen und jeweils die Zahl nacheinander in einem Feld notiert. Erscheint zuerst das Dreiermuster des Zauberers bekommt dieser einen Punkt, sonst der Zuschauer. Dann wird nach diesem erschienenen Dreiermuster weitergemacht und wieder Punkte vergeben. Wer zuerst 5 Punkte hat, hat gewonnen.

Der Trick des Zauberers:

Zu dem gewählten Dreiermuster ABC des Zuschauers wählt der Zauberer das Dreiermuster B'AB, wobei A,B,C eine der Farben S bzw. W sein muß und A', B' und C' die "Gegenfarbe" der Farben A,B,C sind (W ist die "Gegenfarbe" von S und S ist die "Gegenfarbe" von W).

Simulieren Sie diesen Zaubertrick und zeigen, daß die Gewinnwahrscheinlichkeiten für den Zauberer sich wie folgt verhalten:

Zuschauer	WWW	WWS	WSW	WSS	SWW	SWS	SSW	SSS
Zauberer	SWW	SWW	WWS	WWS	SSW	SSW	WSS	WSS
Gewinnchance Zauberer	7/8	3/4	2/3	2/3	2/3	2/3	3/4	7/8

23) Memory

Ein Kartenspiel besteht aus 8 Kartenpaaren (A, A, B, B, C, C, D, D, E, E, F, F, G, G, H, H)

Damit soll das Spiel Memory gespielt werden:

Auf einem 4x4-Spielbrett werden 8 Kartenpaare gelegt, so dass nur ihre Rückseite (Rückseite wird durch O dargestellt) zu sehen ist. Der Spieler darf jeweils zwei Karten aufdecken. Sind die Kartenpaare gleich, so werden diese aus dem Spiel genommen, ungleiche Kartenpaare werden wieder umgedreht.

24) Superhirn (Mastermind)

Folgende Version des Spiels "Superhirn für eine Person" soll implementiert werden:

Der Computer würfelt verdeckt vier Buchstaben aus der Buchstabenmenge:

A, B, C, D, E und F.

Wiederholungen sind dabei erlaubt (Beispiel: ABBA).

Der Spieler soll die Buchstaben und ihre Reihenfolge erraten. Dazu gibt er eine Buchstabenfolge ein

Beispiel: BBDA.

Der Computer antwortet mit X für jeden Buchstaben an der richtigen Stelle und mit z, falls ein Buchstabe in der Folge vorkommt, aber an der falschen Stelle steht

Beispiel: XXz

Die Rateversuche wiederholen sich, bis der Spieler die Buchstabenfolge richtig erraten hat.

Die Anzahl der Rateversuche werden auf dem Bildschirm ausgegeben.

25) Abstände im 2D speichern

Es sind eine bestimmte Menge von Punkten in einem zweidimensionalen Koordinatensystem gegeben. Berechnen Sie die Abstände aller Punkte zueinander und speichern diese in einem zweidimensionalen Feld ab.

26) Duplikate entfernen

In einem Feld aus ganzen Zahlen sollen alle Vielfachen entfernt werden.

Beispiel:

[1, 2, 2, 2, 5, 2, 3, 2, 2, 23] --> [1, 2, 5, 3, 23]

[99, 99, 23, 99, 99, 12, 2] --> [99, 23, 12, 2]

27) Zahlen in ein Feld einfügen

Bemerkung:

In einem Zeichenfeld wird das Ende durch '\0' markiert.

Das geht in einem Feld aus Zahlen nicht.

Vorschlag (Idee):

In einem Feld mit vorgegebener Länge maxLen wird an der Stelle 0 die Position des letzten Element des Feldes (posLastElement) gespeichert!

Wenn das Feld erzeugt wird, ist posLastElement gleich -1.

Nach jedem An - oder Einfügen wird posLastElement um 1 erhöht, bei jedem Löschen eines Elements aus dem Feld wird posLastElement um 1 verringert.

Der Wert von posLastElement darf maximal maxLen-1 sein.

WICHTIG: Die Position des letzten Element des Feldes wird also nicht in einer zusätzlichen Variablen gespeichert, sondern in der Zelle 0 des Feldes!

Beispiele:

Im Feld der Länge 10 werden zu verschiedenen Zeitpunkten verschiedene Anzahl von Zahl gespeichert. Die Belegung kann am Anfang z.B. so aussehen.

Woher soll man wissen, ob z.B. Zahlen von einem Anwender eingegeben wurden und vor allem wie viele es sind bzw. an welcher Stelle sich die letzte Zahl befindet.

-23412	-56723	-34533	-45343	378656	646645	-72354	-34545	98766	178542
--------	--------	--------	--------	--------	--------	--------	--------	-------	--------

Es soll sich keine Zahl im Feld befinden. Dies wird so realisiert:

-1	-56723	-34533	-45343	378656	646645	-72354	-34545	98766	178542
----	--------	--------	--------	--------	--------	--------	--------	-------	--------

Es wird jetzt die Zahl 9 ans Zahlenende angefügt:

1	9	-34533	-45343	378656	646645	-72354	-34545	98766	178542
---	---	--------	--------	--------	--------	--------	--------	-------	--------

Es wird jetzt die Zahl 4 ans Zahlenende angefügt:

2	9	4	-45343	378656	646645	-72354	-34545	98766	178542
---	---	---	--------	--------	--------	--------	--------	-------	--------

Es wird jetzt die Zahl -8 ans Zahlenende angefügt:

3	9	4	-8	378656	646645	-72354	-34545	98766	178542
---	---	---	----	--------	--------	--------	--------	-------	--------

Es wird jetzt die letzte Zahl entfernt

2	9	4	-8	378656	646645	-72354	-34545	98766	178542
---	---	---	----	--------	--------	--------	--------	-------	--------

a)

Schreiben Sie ein Programm, in dem der Anwender immer wieder gefragt wird ob er eine positive Zahl eingeben will (diese wird dann an bisher eingegebenen Zahlen angefügt), ob er eine negative Zahl eingeben will (dann wird die letzte im Feld gespeicherte Zahl entfernt) oder ob er die Zahl 0 eingeben will (dann wird das Programm beendet).

Nach jeder Eingabe wird (zur Kontrolle) der Inhalt des kompletten Feldes auf dem Bildschirm ausgegeben.

28) Noten der Klassenarbeiten archivieren

a) Die Noten einer Klassenarbeit sollen in ein Feld eingelesen werden.

Das Einlesen der Noten wird beendet, wenn eine Zahl außerhalb des Notenbereichs, d.h. wenn als Note ein Wert kleiner 1 oder größer 6, eingegeben wird, oder nachdem das komplette Feld mit Noten belegt ist und es keinen freien Platz mehr hat.

b) Der Mittelwert der Noten soll berechnet werden.

c) Die Noten sollen sortiert (schlechteste Note zuerst) werden.

d) Die sortierten Noten und der Mittelwert sollen auf dem Bildschirm ausgegeben werden.

e) Fügen Sie in dieses sortierte Feld eine Note so ein, daß dieses Feld wieder sortiert ist.

29) Rätsel mit Zahlensystemen

Eine natürliche Zahl n erfüllt die folgenden Bedingungen (Stellen werden von links gezählt):

B1: n hat im Zweiersystem 10 Stellen.

B2: n hat im Fünfersystem 4 Stellen.

B3: Schreibt man n im Zehnersystem, so steht an der letzten Stelle die Ziffer 2.

B4: Schreibt man n im Dreiersystem, so steht an der zweiten Stelle die Ziffer 1.

B5: Schreibt man n im Vierersystem, so steht an der zweiten Stelle die Ziffer 0.

Berechnen Sie n .

30) Rätsel mit Zahlen

Wie viele dreistellige Zahlen gibt es, bei denen die mittlere Ziffer der Mittelwert der ersten und letzten Ziffer ist? Geben Sie diese Zahlen auf dem Bildschirm aus.

Zum Beispiel sind 531 und 420 solche Zahlen.

31) Rätsel um die Jahreszahl 2016 (Rätsel der Woche 3.1.16)

Die Summe aus der Zahl, ihrer Quersumme und der Quersumme ihrer Quersumme ergibt genau 2016. Für welche 4 Zahlen gilt dies ?

32) Neuronales GPS

Einige Insekten und andere Gliedertiere sind mit einem erstaunlich effektiven Navigationssystem ausgestattet. Ihr Nervensystem zeichnet die Geschwindigkeit und Richtung ihrer Bewegung relativ zu einem Ausgangspunkt auf.

Anhand dieser Informationen gelingt es den Tieren z.B. einen direkteren und kürzeren Rückweg zu finden (Spektrum der Wissenschaften 10/16)

a)

1. Modellierung der Daten des Lebewesens:

Das Lebewesen bewegt sich in einem bestimmten Winkel α eine bestimmte Zeitdauer Δt (in Sekunden s) mit einer bestimmten Geschwindigkeit v (in m/s).

Berechnen Sie den Wert der x- und y-Koordinate nach n Zeitabschnitten

Beispiel:

Zeitabschnitte	1	2	3	4	5	6
Δt (s)	2	1	3	2	1	3
v (m/s)	4	-2	5	4	-2	5
Winkel ($^\circ$)	30	189	290	30	189	290

b)

2. Modellierung der Daten des Lebewesens:

Im 1. Zeitabschnitt bewegt es sich in einem bestimmten Winkel α eine bestimmte Zeitdauer Δt (in Sekunden s) mit einer bestimmten Geschwindigkeit v (in m/s), so wie oben bei a)

Danach wird der Winkel α angegeben, mit dem es sich relativ zur vorigen Richtung weiterbewegt.

Berechnen Sie den Wert der x- und y-Koordinate nach n Zeitabschnitten

Beispiel:

Zeitabschnitte	1	2	3	4	5	6
Δt (s)	2	1	3	2	1	3
v (m/s)	4	2	5	4	-2	5
Winkel ($^\circ$)	30	40	290	30	189	290

Nach dem 2. Zeitabschnitt bewegt sich das Lebewesen relativ zur vorigen Richtung um 40° .

Der Gesamtwinkel beträgt jetzt also $30^\circ + 40^\circ = 70^\circ$

33) Parser

Ein Parser stellt fest, ob die vom Scanner überreichten Eingabesymbole (Wörter) ein korrektes Konstrukt ergeben. Zum Beispiel kann er nachprüfen, ob eine Zeichenfolge ein (aus der Mathematik bekannter) arithmetischer Term ist, wie z.B. $((23+7)-4)$

33.1) Einfachster geklammerter Term

Ein "einfachster Additionsterm" soll hier eine Zeichenfolge sein, die aus dem Kleinbuchstaben a , dem Rechenzeichen $+$ besteht und geklammert ist (außer die Zeichenfolge besteht genau aus einem Kleinbuchstaben a).

Beispiele für "geklammerte Terme":

a $(a+a)$ $(a+(a+a))$ $((a+a)+(a+a))$ $((a+(a+a))+((a+a)+(a+a)))$

Beispiele für keine "geklammerten Terme" (fehlgeformt):

r $(a$ (a) $(a++a)$ $(a+a)+a$

Aufgabe:

Erstellen Sie ein Programm, das von einem Array von Zeichen (Zeichenkette) berechnet, ob es ein "einfachster Additionsterm" ist oder nicht.

Tipp 1:

Wenn man sich verschiedene Terme anschaut, dann gibt es 4 Fälle:

Fall1: Einer Klammer (muß eine Klammer (oder eine Ziffer folgen). Kurzschreibweise:

Fall1 $(\implies (, +$

Fall2: $) \implies) , +$

Fall3: $+ \implies (, \text{Ziffer}$

Fall4: $\text{Ziffer} \implies) , +$

Ansonsten ist es kein Term. Außerdem muß bei einem Term die Anzahl der öffnenden Klammern gleich der Anzahl der schliessenden Klammern sein.

Ist der zu dem Tipp zugehörige Algorithmus (ist eine Zeichenfolge ein Term?) korrekt?

Implementieren Sie den Algorithmus und testen Sie ausführlich!

Tippe 2:

Ersetzen Sie in der Zeichenkette alle Ausdrücke der Form $(a+a)$ durch a .

Beispiel:

Die am weitesten rechts stehende, öffnende Klammer wird hier durch die Klammer $[$ kenntlich gemacht. Danach kommt die Ersetzung.

Zu prüfende Zeichenkette: $(((((a+a) + a) + (a+a)) + a)$

1. Ersetzung: $((((a+a) + a) + a)$

2. Ersetzung: $(((a+a) + a)$

3. Ersetzung: $(a+a)$

4. Ersetzung: a

33.2) geklammerter Variablenterm

Ein "geklammerter Variablenterm" soll hier eine Zeichenfolge sein, die aus Kleinbuchstaben a - z, den Rechenzeichen +, -, *, / besteht und geklammert ist (außer die Zeichenfolge besteht genau aus einem Kleinbuchstaben, also a - z).

Beispiele für "geklammerte Terme":

b	(x*y)
((e*(f*f))	((a+b)+(c-d))

Beispiele für keine "geklammerten Terme" (fehlgeformt):

A	(x)
(a--b)	(a*/b)+c

Aufgabe:

Erstellen Sie ein Programm, das von einem Array von Zeichen (Zeichenkette) berechnet, ob es ein "geklammerter Variablenterm" ist oder nicht.

33.3) Zahlenterm

Ein "Zahlenterm" soll hier eine Zeichenfolge sein, die aus den Ziffern 0 bis 9, den Rechenzeichen +, -, *, / besteht und geklammert ist (außer die Zeichenfolge besteht genau aus einer Ziffer).

Beispiele für Zahlenterme:

7	(2+3)	(((2*3)+4)-(8+9))
---	-------	-------------------

mit den Werten:

7	5	-7
---	---	----

Beispiele für keine Beispiele für Zahlenterme:

a	(5	(5+-6)	((6+3)))
---	----	--------	----------

Aufgabe:

Erstellen Sie ein Programm, das von einem Array von Zeichen (Zeichenkette) berechnet, ob es ein "Zahlenterm" ist. Wenn ja, soll auch der Wert des Zahlenterm berechnet werden.

33.4) gemischter Zahlenterm

Ein "gemischter Zahlenterm" ist ein Zahlenterm (siehe letzte Aufgabe), in dem auch ungeklammerte Ausdrücke (ähnlich dem aus der Mathematik bekanntem arithmetischen Term) vorkommen dürfen und wo die Regel "Punkt vor Strich" gilt.

Beispiele für Zahlenterme:

7	2+3*4	((2*3)+4)-8+9
---	-------	---------------

Aufgabe:

Erstellen Sie ein Programm, das von einem Array von Zeichen (Zeichenkette) berechnet, ob es ein "gemischter Zahlenterm" ist. Wenn ja, soll auch der Wert des Zahlenterm berechnet werden.

C-ÜBUNGSAUFGABEN FELDER 2

34) Unerlaubter Speicherzugriff

Demonstration unerlaubter Speicherzugriff (auf nicht reservierten Speicher):

a) Lassen Sie das Programm (siehe unten) auf einem Rechner laufen.

Erklären Sie die merkwürdige Ausgabe auf dem Bildschirm.

symbolische Adresse	Inhalt
....	
f2[0]	T
f2[1]	O
f2[2]	M
f2[3]	'\0'
f1[0]	U
f1[1]	L
f1[2]	F
f1[3]	'\0'
...	

Beispiel:

```
int main() {
    char f1[4]="ulf";
    char f2[4]="tom";

    f2[4]='a';
    f2[5]='s';
    f2[6]='s';

    printf("f1=%s\n", f1);
    printf("f2=%s\n", f2);

    return 0;
}
```

35)

Die mit einem Würfel erzielten Ergebnisse werden in einem Feld gespeichert.

Schreiben Sie ein Programm, das folgendes berechnet und dann auf dem Bildschirm ausgibt:

Die relative Häufigkeit (z.B. Anzahl der 6-en / Gesamtzahl aller Würfe) der Zahl 6.

Die relative Häufigkeit aller anderen Zahlen 1, 2, 3, 4, 5.

Die relative Häufigkeit zweier hintereinander gewürfelter 1-er, 2-er, ..., 6-er.

Der theoretische Wert (Erwartungswert) soll jeweils (zum Vergleich) auch noch auf dem Bildschirm ausgegeben werden.

Ist die Wahrscheinlichkeit eine weitere 6 zu würfeln auch noch $1/6$, wenn vorher fünf 6-en gewürfelt wurden ?

36)

Ein Konzern besteht aus 3 Unternehmen mit je 4 Filialen.

a) Schreiben Sie ein Programm, mit dem man für jedes Unternehmen und jede Filiale die Anzahl der Mitarbeiter einlesen kann.

b) Erweitern Sie das Programm so, dass es die Gesamtanzahl der Mitarbeiter des Konzerns berechnet und ausgibt.

c) Erweitern Sie das Programm so, dass es die Gesamtanzahl der Mitarbeiter pro Unternehmen berechnet und ausgibt.

37)

Die Buchhaltung benötigt ein Programm, das folgendes berechnet (und alles in **einem** Feld speichert):

- Gesamter Quartalsumsatz für jede Warengruppe
- Gesamter Warenumsatz für jeden Monat
- Gesamtumsatz.

Testen Sie Ihr Programm mit folgenden Werten:

Warengruppen	A	B	C	D	E	Summen
Januar	3245	55654	356	6765	377	
Februar	7657	653	6765	7665	6753	
März	9872	1275	8977	1234	566	
Summen						

38)

Erstellen Sie ein Programm, das in der letzten Spalte einer Tabelle die jeweilige Zeilensumme und in der letzten Zeile der Tabelle die jeweilige Spaltensumme abspeichert

In dem letzten rechten unteren Element der Tabelle muss die Summe aller Tabellenelemente (außer den Elementen der letzten Zeile und der letzten Spalte) berechnet und abgespeichert werden. (siehe vorige Aufgabe).

Wichtig:

Die Zeilenanzahl ZANZ und die Spaltenanzahl SANZ müssen Konstanten sein, die der Programmierer verändern kann.

39)

Gegeben sei ein zweidimensionales Feld.

Schreiben Sie ein Programm, das das Feld analysiert und folgende Resultate anzeigt:

- Maximum des Feldes
- Minimum des Feldes
- Anzahl der geraden Zahlen des Feldes
- Anzahl der ungeraden Zahlen des Feldes
- Zeilensummen
- Spaltensummen

40)

Schreiben Sie ein Programm, das in dem Feld

jahr[monat][tag]

jeden Wochentag des Jahres 2003 abspeichert.

Der Anwender kann dann z.B. erfahren was für ein Wochentag z.B. der 27.6.2003 ist.

Testen Sie das Programm.

41)

a) Vertauschen Sie die Zeilen und Spalten (also 1. Zeile mit 1. Spalte, 2. Zeile mit 2. Spalte, usw.) eines zweidimensionalen Feldes, das gleich viel Zeilen und Spalten besitzt.

b) Diskutieren Sie folgenden Lösungsvorschlag (dynamische Entwicklung der Variablen in einer Tabelle festhalten)

```
...
for (i=0; i<len; i++) {
    for (j=0; j<len; j++) {
        temp=v[i][j];
        v[i][j] = v[j][i];
        v[j][i]=temp;
    }
}
```

c) Diskutieren Sie folgenden Lösungsvorschlag (dynamische Entwicklung der Variablen in einer Tabelle festhalten)

```
...
for (i=0; i<len; i++) {
    for (j=i; j<len; j++) {
        temp=v[i][j];
        v[i][j] = v[j][i];
        v[j][i]=temp;
    }
}
```

42)

a) Vertauschen Sie eine Zeile eines 2-dimensionalen Feldes mit einer anderen Zeile.

b) Vertauschen Sie eine Spalte eines 2-dimensionalen Feldes mit einer anderen Spalte.

Der Anwender soll eingeben, welche Zeilen bzw. Spalten dies sind.

C-ÜBUNGSAUFGABEN FELDER 3

43)

Das Sieb des Eratosthenes:

Vorbemerkung: Eine Zahl ist eine Primzahl, wenn sie nur durch 1 und sich selbst teilbar ist.

D.h. 2, 3, 5, 7, 11, 13, 17, 19, 23, ... sind Primzahlen.

Es sollen alle Primzahlen bis zu einer vom Anwender einzugebenden Zahl n ausgegeben werden. Hierzu soll das " Sieb des Eratosthenes" verwendet werden.

Zunächst werden alle Zahlen von 1 bis n aufgeschrieben und mit einer 0 markiert:

0	0	0	0	0	0	0	0	0	0	0	0	0	...					0
1	2	3	4	5	6	7	8	9	10	11	12	...						n

Zuerst wird i auf 2 gesetzt.

Nun testet man, ob i , also 2, eine Primzahl ist. Da dies der Fall ist, wird i , also 2, auf dem Bildschirm ausgegeben. Anschließend werden alle Vielfache von i , also 4, 6, 8, 10, ... mit einer 1 markiert.

0	0	0	1	0	1	0	1	0	1	0	1	...						
1	2	3	4	5	6	7	8	9	10	11	12	...						n

Dann wird i auf 3 gesetzt.

Nun testet man, ob i , also 3 eine Primzahl ist. Da dies der Fall ist, wird i , also 3, auf dem Bildschirm ausgegeben. Anschließend werden alle Vielfache von i , also 6, 9, 12, ... mit einer 1 markiert.

0	0	0	1	0	1	0	1	1	1	0	1	...						
1	2	3	4	5	6	7	8	9	10	11	12	...						n

Dann wird i auf 4 gesetzt.

Nun testet man, ob i , also 4 eine Primzahl ist. Da dies nicht der Fall ist, geschieht nichts.

Dann wird i auf 5 gesetzt.

Usw.

44) Primzahlenliste

Eine Zahl $n > 2$ ist eine Primzahl, genau dann wenn alle kleineren Primzahlen (als n) sie nicht teilen. Damit kann man eine Primzahl durch die Liste ihrer kleineren Primzahlen entwickeln. Die Liste beginnt mit der kleinsten Primzahl $L = [2]$.

Um von einer Zahl $n \geq 3$ zu bestimmen, ob sie eine Primzahl ist macht man Folgendes (hier am Beispiel $n=14$):

x1) Beginne mit 3

2 teilt nicht 3 \implies Listenende erreicht \implies füge 3 an L an, ergibt neues $L = [2,3]$

x2) Mache mit 4 weiter:

2 teilt 4 \implies Ende

x3) Mache mit 5 weiter:

2 teilt nicht 5 \implies mache mit nächstem Element von L (also 3) weiter

3 teilt nicht 5 \implies Listenende erreicht \implies füge 5 an L an, ergibt neues $L = [2,3,5]$

x4) Mache mit 6 weiter:

2 teilt 6 \implies Ende

x5) Mache mit 7 weiter:

2 teilt nicht 7 \implies mache mit nächstem Element von L (also 3) weiter

3 teilt nicht 7 \implies mache mit nächstem Element von L (also 5) weiter

5 teilt nicht 7 \implies Listenende erreicht \implies füge 7 an L an, ergibt neues $L = [2,3,5,7]$

x6) Mache mit 8 weiter:

2 teilt 8 \implies Ende

x7) Mache mit 9 weiter:

2 teilt nicht 9 \implies mache mit nächstem Element von L (also 3) weiter

3 teilt 9 \implies Ende

x8) Mache mit 10 weiter:

2 teilt 10 \implies Ende

x9) Mache mit 11 weiter:

2 teilt nicht 11 \implies mache mit nächstem Element von L (also 3) weiter

3 teilt nicht 11 \implies mache mit nächstem Element von L (also 5) weiter

5 teilt nicht 11 \implies mache mit nächstem Element von L (also 7) weiter

7 teilt nicht 11 \implies Listenende erreicht \implies füge 11 an L an, ergibt neues $L = [2,3,5,7,11]$

x10) Mache mit 12 weiter:

2 teilt 12 \implies Ende

x11) Mache mit 13 weiter:

2 teilt nicht 13 \implies mache mit nächstem Element von L (also 3) weiter

3 teilt nicht 13 \implies mache mit nächstem Element von L (also 5) weiter

5 teilt nicht 13 \implies mache mit nächstem Element von L (also 7) weiter

7 teilt nicht 13 \implies mache mit nächstem Element von L (also 11) weiter

11 teilt nicht 13 \implies Listenende erreicht \implies füge 11 an L an, ergibt neues $L = [2,3,5,7,11,13]$

x12) Mache mit 14 weiter:

2 teilt 14 \implies Ende

45) Das Sieb der glücklichen Zahlen

Es sollen alle "glücklichen" Zahlen bis zu einer vom Anwender einzugebenden Zahl n ausgegeben werden. Hierzu soll ein "Sieb" verwendet werden (nicht ausgesiebte Zahlen nennt man glücklich).

Zunächst werden alle natürliche Zahlen von 1 bis n aufgeschrieben und die 1 markiert.

Beispiel $n=20$

Die nächste nicht ausgesiebte Zahl ist 2, also wird jede 2. Zahl gestrichen:

	#																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

Das ergibt dann:

		#																	
1		3		5		7		9		11		13		15		17		19	
21		23		25		27		29		31		33		35		37		39	

Die nächste nicht ausgesiebte Zahl ist 3, also wird jede 3. Zahl gestrichen:

Das ergibt dann:

						#													
1		3				7		9				13		15				19	
21				25		27				31		33				37		39	

Die nächste nicht ausgesiebte Zahl ist 7, also wird jede 7. Zahl gestrichen:

Das ergibt dann:

1		3				7		9				13		15					
21				25		27				31		33				37			

Die nächste nicht ausgesiebte Zahl ist 9, also wird jede 9. Zahl gestrichen:

Das ergibt dann:

1		3				7		9				13		15					
21				25						31		33				37			

usw.

Das ergibt die Folge der glücklichen Zahlen:

1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, 87, 93, 99, 105, 111, 115, 127, 129, 133, 135, 141, 151, 159, 163, 169, 171, 189, 193, 195, 201, 205, 211, 219, 223, 231, 235, 237, 241, 259, 261, 267, 273, 283, 285, 289, 297, 303, ...

46.1)

Um ein 2-dimensionales Feldes zu kodieren wird folgendes gemacht:

Die 1. Spalte wird mit der letzten vertauscht.

Die 2. Spalte wird mit der 2. letzten vertauscht, usw.

46.2)

Um ein 2-dimensionales Feldes zu kodieren wird folgendes gemacht:

Die 1. Zeile wird mit der letzten vertauscht.

Die 2. Zeile wird mit der 2. letzten vertauscht, usw.

47)

Ersetzen Sie in einer Zeichenkette (die z.B. ein Anwender eingibt), alle Umlaute "ÄÖÜäöüß" durch entsprechende Doppelbuchstaben "AeOeUeaeoeuess".

48)

Vertauschen Sie in einer Zeichenkette die Reihenfolge der Zeichen:

Aus z.B. "rot" wird "tor"

49) Kryptologie (RSA-Verfahren)

Mit dem RSA-Verfahren (Rivest, Shamir, Adleman) kann man über ein prinzipiell abhörbares Netz trotzdem einen geheimen Nachrichtenaustausch organisieren.

Dazu trägt in einer frei zugänglichen Schlüsselbibliothek (ähnlich einem Telefonbuch) der Teilnehmer A einen so genannten öffentlichen Schlüssel (k, n) , d.h. ein Zahlenpaar ein.

Zusätzlich besitzt A noch einen so genannten privaten Schlüssel (k', n) , den nur er kennt. (n ist öffentlich und daher allen bekannt).

Ein Teilnehmer B, der eine Nachricht m an A senden will, chiffriert (verschlüsselt) m mit Hilfe von k zum so genannten Chiffre c und versendet dann nicht m , sondern das verschlüsselte c an A.

A dechiffriert dann c mit Hilfe von k' wieder zu m . A und B haben also verschiedene Aufgaben:

Aufgabe von B: $m \xrightarrow{k} c$

Aufgabe von A: $c \xrightarrow{k'} m$

Mit Hilfe des RSA-Verfahrens kann man eine große Menge von Schlüsselpaaren (k, k') für viele potentielle Teilnehmer erzeugen.

Die Erzeugung eines konkreten Schlüsselpaares (k, k') erfolgt am besten vor Ort (d.h. bei A), damit nicht noch ein zusätzliches Schlüsselaustauschproblem entsteht.

Hier das Verfahren an einem Beispiel:

(k, n) : öffentliche Schlüssel (natürliche Zahlen > 0)

(k', n) : geheimer Schlüssel (natürliche Zahlen > 0)

$m < n$: zu übermittelnde (noch unverschlüsselte) Nachricht

c : die Chiffre (mit k verschlüsselte Nachricht m)

d : die dechiffrierte (entschlüsselte) Chiffre c

Bemerkung zum Operator mod:

mod berechnet den Rest, der bei einer Division übrig bleibt.

Beispiele: $9 \bmod 5 = 4$, $8 \bmod 2 = 0$, $7 \bmod 2 = 1$

Folgende Schritte müssen durchgeführt werden, um Schlüsselpaars (k, k') zu erzeugen:

Schritt 1:

wähle 2 Primzahlen $p > 1$ und $q > 1$ mit $p \neq q$ und bilde das Produkt $n = p \cdot q$

wähle zum Beispiel: $p = 11$ und $q = 31$ mit

$$n = p \cdot q = 11 \cdot 31 = 341$$

Schritt 2:

Berechne das Produkt r mit

$$r = (p-1) \cdot (q-1) = 10 \cdot 30 = 300$$

Schritt 3:

Berechne das Produkt (Schlüsselprodukt) s mit

$s = v \cdot r + 1$ wobei v eine frei wählbare, natürliche Zahl größer gleich 1 ist.

wähle zum Beispiel $v=1$. Dies ergibt:

$$s = 1 \cdot r + 1 = 301$$

Schritt 4:

Wähle k und k' so, daß folgende Bedingungen gelten:

B1) $k \cdot k' = s$ und

B2) $1 < k < r$ und

B3) k und r sind teilerfremd (außer 1)

Dann hat man ein mögliches Schlüsselpaar (k, k') erzeugt.

Da $301 = 7 \cdot 43$ und $k = 7 < r = 300$ und $k = 7$ und $r = 300$ teilerfremd sind, ist $(k, k') = (7, 43)$ ein Schlüsselpaar.

Schritt 5: (Chiffrierung)

Chiffriere (verschlüsse) m wie folgt zu c , wobei $m < n$ sein muß:

$$c = m^k \bmod n$$

wähle zum Beispiel: $m = 2 < n = 341$. Dann gilt:

$$c = m^k \bmod n = 2^7 \bmod 341 = 128$$

Schritt 6:

Teste, ob die Dechiffrierung (Entschlüsselung) korrekt funktioniert, d.h. wiederum m ergibt:

$$d = c^{k'} \bmod n = 128^{43} \bmod 341 = 2$$

Bemerkung:

Das RSA-Verfahren war in den USA von 1983 bis 2000 patentiert.

Weitere Infos im Buch von : Prof. Uwe Schöning "Ideen der Informatik"

Aufgabe:

Schreiben Sie ein Programm, das mehrere Schlüsselpaare (k, k') erzeugt und in einem Feld speichert.

C-ÜBUNGSAUFGABEN FELDER 4

50)

Mit einem Würfelbecher, in dem sich drei Würfel (W1, W2, W3) befinden, werden n (z.B. n = 120) Würfe gemacht. Dies muss in einem Programm simuliert werden. Ein Statistiker interessiert sich für die Eigenschaften der Augenzahlen (1 bis 6), der Würfe (1 bis n, der Würfel (W1, W2, W3 und alle 3 Würfe gemeinsam)

Beispiel: 3 Würfel, 12 Würfe.

													Summe	Mittelw.
W1	4	2	1	6	6	3	1	5	4	2	3	2	39	3,25
W2	2	2	1	6	1	2	2	1	5	1	5	5	33	2,75
W3	5	2	3	6	5	3	4	2	6	5	1	6	48	4
Summe	11	6	5	18	12	8	7	8	15	8	9	13		
Mittelw.	3,67	2	1,67	6	4	2,67	2,33	2,67	5	2,67	3	4,33		

Auge		1	2	3	4	5	6
W1	absolut	2	3	2	2	1	2
	relativ	1/6	1/4	1/6	1/6	1/12	1/6
W2	absolut	4	4	0	0	3	1
	relativ	1/3	1/3	0	0	1/4	1/12
W3	absolut	1	2	2	1	3	3
	relativ	1/12	1/6	1/6	1/12	1/4	1/4

- a) Speichern Sie die Würfe und Auswertungen (absolute, relative Häufigkeiten, Mittelwerte usw.) in entsprechenden Feldern.
- b) Geben Sie die Auswertungen (absolute, relative Häufigkeiten, Mittelwerte usw.) auf dem Bildschirm aus.
- c) Ermitteln Sie mittels der Auswertungen, die relative Häufigkeit für das Vorkommen von drei Sechsen in einem Wurf. Erweitern Sie die Anzahl der Würfe von n=120 auf z.B. n=12000, damit Sie ein genaueres Ergebnis bekommen.
- d) Speichern Sie die Würfe und Auswertungen in einer Datei.

Einfaches Demo-Programm zur Erzeugung einer Zufallszahl

```
#include "stdafx.h"    // Einfachstes Beispielprogramm:
#include <stdlib.h>      // Siehe auch Hilfe-Menü zu VC++
#include "stdio.h"      // Erzeugt eine Zufallszahl
#include <time.h>

void main(){
    int wuerfel;
    // damit bei jedem Programmaufruf nicht die gleiche Zufallszahl erzeugt wird
    // am Programmbeginn genau ein Mal srand(...) aufrufen!!
    srand((unsigned)time( NULL ));
    wuerfel = rand()%6+1; // erzeugt eine Zufallszahl
}
```

51) Geburtstagsparadoxon

In einem Raum befinden sich 23 Personen. Wie groß ist die Wahrscheinlichkeit, daß zwei oder mehr dieser Personen am gleichen Tag (ohne Beachtung des Jahrganges) Geburtstag haben? (Ein Jahr hat 365 Tage).

a) Schätzen Sie diese Wahrscheinlichkeit.

b) Simulieren Sie dieses Problem und vergleichen das Ergebnis mit Ihrer Schätzung.

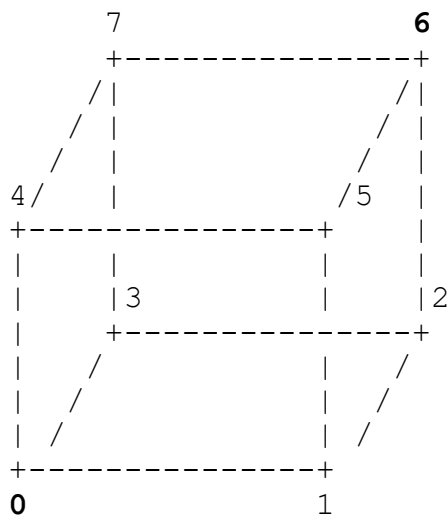
52) Ameisenwürfel (Martin Gardner)

Eine Ameise beginnt an der Ecke 0 eines Würfels loszulaufen.

Mit jeweils der Wahrscheinlichkeit $1/3$ läuft sie zu einer der 3 benachbarten Ecken.

Wenn sie zur Ecke 6 kommt, ist ihre Reise beendet, da dort Honig klebt.

Stimmt es, dass eine Ameise im Durchschnitt 10 Kanten weit wandern muss, bis sie zur Ecke 6 kommt?



53) Lotto 1

Schreiben Sie ein Programm, das die Auspielungen (Ziehungen) des Glücksspiels Lotto (6 verschiedene Zahlen aus 49) simuliert:

a) Der Anwender soll seinen Lotto-Tip ("Dauertip") über Tastatur eingeben.

b) Das Programm soll eine Auspielung (6 verschiedene Zahlen) erzeugen und auf dem Bildschirm ausgeben.

c) Das Programm soll 10 Auspielungen erzeugen und dem Anwender für jede Auspielung die Anzahl der Richtigen (verglichen mit dem Dauertip) ausgeben.

d) Das Programm soll zählen wie oft Sie 0 Richtige, 1 Richtige, ..., 6 Richtige haben. Die einzelnen Auspielungen sollen nicht ausgegeben werden, sondern nur die Häufigkeiten für 0 bis 6 Richtige. Das Programm soll 520 Auspielungen simulieren (das entspricht der Anzahl der Auspielungen in zehn Jahren bei einem wöchentlichen Lotto-Tip).

54) Lotto 2

Schreiben Sie ein Programm, das die Ausspielungen (Ziehungen) des Glücksspiels Lotto (6 verschiedene Zahlen aus 49) simuliert:

- a) Der Anwender soll seinen Lotto-Tip ("Dauertip") über Tastatur eingeben.
- b) Das Programm soll eine Ausspielung (6 verschiedene Zahlen) erzeugen und auf dem Bildschirm ausgeben.
- c) Das Programm soll 10 Ausspielungen erzeugen und dem Anwender für jede Ausspielung die Anzahl der Richtigen (verglichen mit dem Dauertip) ausgeben.
- d) Das Programm soll zählen wie oft Sie 0 Richtige, 1 Richtige, ..., 6 Richtige haben. Die einzelnen Ausspielungen sollen nicht ausgegeben werden, sondern nur die Häufigkeiten für 0 bis 6 Richtige. Das Programm soll 520 Ausspielungen simulieren (das entspricht der Anzahl der Ausspielungen in zehn Jahren bei einem wöchentlichen Lotto-Tip).

55)

55.1) Monte-Carlo-Methoden

- a) Jemand geht mit 100 Euro Anfangskapital ins Spielcasino und setzt immer wieder 10 Euro auf die Farbe rot. Das macht er so lange, bis er entweder 110 Euro oder 0 Euro Endkapital hat, d.h. bis er insgesamt 10 Euro gewonnen oder die ganzen 100 Euro Anfangskapital verloren (verspielt) hat. Dann verlässt er das Spielcasino.

Simulieren Sie diese Strategie durch ein C-Programm.

- b) Um herauszufinden, wie groß der durchschnittliche (mittlere) Gewinn mit dieser Strategie ist, geht Herr X insgesamt $n = 100000$ Mal ins Spielcasino und kann damit seinen durchschnittlichen Gewinn berechnen. Da Herr X dann bis dahin sehr alt wäre, beauftragt er Sie, ein geeignetes C-Programm zu schreiben, das diesen Sachverhalt simuliert.

Konkret:

Simulieren Sie diese bei a) eingesetzte Strategie $n = 100000$ durch ein geeignetes C-Programm und geben den durchschnittlichen Gewinn dieser Strategie auf dem Bildschirm aus.

55.2) Monte-Carlo-Methoden

Jemand benutzt im Spielcasino die folgende Strategie:

Er setzt seinen Einsatz von 1 Euro auf die Farbe "rot".

Erscheint die Farbe "rot", hört er sofort auf und er hat 1 Euro gewonnen.

Erscheint die Farbe "schwarz" (dann hat er bis jetzt 1 Euro verloren), setzt er wieder auf "rot" und verdoppelt seinen Einsatz auf 2 Euro.

Erscheint die Farbe "rot", hört er sofort auf und er hat $2 \text{ Euro} - 1 \text{ Euro} = 1 \text{ Euro}$ gewonnen.

Erscheint die Farbe "schwarz" (dann hat er bis jetzt 3 Euro verloren), setzt er wieder auf "rot" und verdoppelt seinen Einsatz auf 4 Euro.

usw.

Dies macht er aber insgesamt maximal 10 mal.

Mit welcher Wahrscheinlichkeit ist sein Gewinn 10 Euro ?

Simulieren Sie diese Strategie $n = 1000$ Mal am Rechner und berechnen den durchschnittlichen (mittleren) Gewinn.

C-ÜBUNGSAUFGABEN FELDER 5

56) Scanner (oder Tokenizer)

Ein sogenannter Scanner ist Teil eines Compilers. Er zerlegt eine Zeichenkette in Tokens (lexikalische Einheiten, Atome).

Das sind hier konkret: Die Rechenzeichen -, +, *, / und Zahlen. Eine Zahl darf auch eine Gleitkommazahl wie -12.45 sein. Eine Zahl darf dagegen nicht mit einem Punkt enden.

Deshalb ist z.B. die Zeichenkette 34. keine Zahl, sondern syntaktisch falsch.

Kommt in der Zeichenkette ein Zeichen vor, das keine Ziffer, kein Punkt bzw. kein Rechenzeichen ist, dann ist diese Zeichenkette auch syntaktisch falsch.

a) Es wird vorausgesetzt, daß der Anwender jeweils die Tokens durch die Eingabe eines Leerzeichens trennt.

Die folgende Zeichenkette (gespeichert in einem 1-dimensionalen Feld)

34	-	*	2.23	+	9
----	---	---	------	---	---

wird in die folgende Tokens zerteilt (gespeichert in einem 2-dimensionalen Feld) :

34	-	*	2.23	+	9
----	---	---	------	---	---

Schreiben Sie dazu das entsprechende Programm.

Geben Sie die Werte aller Tokens auf dem Bildschirm aus.

Das Programm soll außerdem feststellen, ob sich ein fehlerhaftes Token in der Zeichenkette befindet.

b) Es wird vorausgesetzt, daß der Anwender jeweils die Tokens durch die Eingabe eines oder mehrerer Leerzeichen trennt.

Schreiben Sie dazu das entsprechende Programm.

c) Es wird vorausgesetzt, daß der Anwender die Tokens durch die Eingabe eines oder mehrerer Leerzeichen trennen kann, aber nicht muß..

Die folgende Zeichenkette

34-+2.23+9

wird also in die folgende Tokens zerteilt:

34	-	*	2.23	+	9
----	---	---	------	---	---

Um später schneller die Tokens identifizieren zu können, müssen sie vorne mit einer Abkürzung markiert werden.

Z: ziffer

R: Rechenzeichen

Z34	R-	R*	Z2.23	R+	R9
-----	----	----	-------	----	----

```
// Demoprogramm: Umwandeln Zeichenkette in Zahl
#include "stdafx.h"
#include "stdio.h"
#include "stdlib.h"

int main(){
    double d;
    int i1;
    int i2;

    char str1[10]="3.1415";
    char str2[10]="1234";
    char str3[10]="Z5678";

    // Umwandeln einer Zeichenkette in eine double-Zahl
    d = atof(str1);
    // Umwandeln einer Zeichenkette in eine integer-Zahl
    i1 = atoi(str2);
    // Umwandeln einer Zeichenkette in eine integer-Zahl,
    // aber erst ab dem 1. Zeichen, also ab der Ziffer 5
    i2 = atoi(&str3[1]);

    printf("d=%f\n",d);
    printf("i1=%d\n",i1);
    printf("i1=%d\n",i2);

    return 0;
}
```

C-ÜBUNGSAUFGABEN FELDER 6

57) Mathematische Probleme

57.1) Arithmetische Progression

Wie viele Pluszeichen muss man zwischen die Ziffern der Zahl 123456789 setzen, damit als Summe 99 herauskommt?

57.2) Arithmetische Progression

Bemerkungen:

Eine arithmetische Progression der Länge n mit Abstand d sind n aufeinanderfolgende Zahlen mit Abstand d .

Beispiel:

3, 7, 11, 15, 19 ist eine arithmetische Progression der Länge 5 mit Abstand 4.

Frage:

Kann man die Menge $\{1; 2; 3; 4; 5; 6; 7; 8; 9\}$ so in 2 Teilmengen zerlegen, daß keine von ihnen eine arithmetische Progression enthält ?

Quelle: Spektrum der Wissenschaft Mai 2012 Seite 23

57.3) Diskrepanz-Problem von Paul Erdős

Siehe Spektrum der Wissenschaft Januar 2016.

Ein Gefangener befindet sich auf der Mitte eines Berggipfels, der $b = 3$ Schritte breit ist, wobei sich links und rechts dieses Berggipfels jeweils ein tiefer Abgrund befindet (siehe Skizze unten).

Um den Gefangenen zu quälen, zwingt ihn sein Wärter, sich nach links bzw. rechts zu bewegen.

Kann der Gefangene insgesamt eine bestimmte vorgegebene Anzahl n (z.B. $n = 8$) von Schritten machen, ohne in einen Abgrund zu stürzen?

Die Lösung wäre, wenn der Gefangene abwechselnd in beide Richtungen geht (z.B. links, rechts, links, rechts, usw.) Allerdings hat sich der Wärter noch eine Gemeinheit einfallen lassen:

Der Wärter kann noch bestimmen, ob der Gefangene nur jeden zweiten Schritt (beginnend mit dem zweiten) ausführen darf. Oder er läßt nur jeden 3.-ten, 4.-ten, ..., k -ten Schritt zu.

Frage:

Existiert eine Schrittfolge für die n Schritte, so daß der Gefangene nicht in den Abgrund stürzt, unabhängig, welche Strategie der Wärter wählt ?

Beispiel: (-1 bedeutet einen Schritt nach links, 1 bedeutet einen Schritt nach rechts)

-1, 1, 1, -1, -1, 1, 1, -1,

Strategie 1 des Wärters: Jeder Schritt ($k=1$):

-1, 1, 1, -1, -1, 1, 1, -1

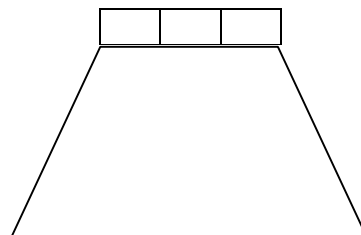
Ergebnis: Gefangener überlebt.

Strategie 2 des Wärters: Jeder 2. Schritt ($k=2$):

-1, 1, 1, -1, -1, 1, 1, -1

also: 1, -1, 1, -1

Ergebnis: Gefangener überlebt.



Strategie 3 des Wärters: Jeder 3. Schritt ($k=3$):

-1, 1, **1**, -1, -1, **1**, 1, -1

also:

1, 1

Das ist ein Absturz!

Ergebnis: Gefangener überlebt nicht.

Ergebnis:

Bei dieser Zugfolge wählt der Wärter die Strategie 3 und gewinnt damit das Spiel.

Aufgaben:

a) Zeigen Sie (z.B. mit brute force) , daß der Gefangene eine Schrittfolge der Anzahl $n = 11$ angeben kann, ohne daß der Wärter eine Strategie findet, so daß der Gefangene abstürzt.

b) Gilt das auch für $n = 12$?

Bemerkung:

Das Problem kann auch für Bergipfel mit anderen Breiten ($b = 5, 7, 9, \dots$) untersucht werden.

57.4) Josephus-Problem

Von $n = 13$ Schülern einer Klasse soll einer in einem Glücksspiel gewinnen.

Die n Schüler bilden einen Kreis, wobei jeder 3. Schüler den Kreis verlassen muß. Wer übrig bleibt, hat gewonnen.

Anders formuliert (kurz gefasst):

13 Leute stehen durchnummeriert im Kreis. Nacheinander muss jeder Dritte gehen.

Welche Personen bleiben übrig?

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

Elimination jedes 3. Schülers ergibt:

1	2		4	5		7	8		10	11		13
---	---	--	---	---	--	---	---	--	----	----	--	----

Elimination jedes 3. Schülers ergibt (nach 13 folgt im Kreis die 1)

1			4	5			8		10			13
---	--	--	---	---	--	--	---	--	----	--	--	----

Elimination jedes 3. Schülers ergibt

1				5			8					13
---	--	--	--	---	--	--	---	--	--	--	--	----

Elimination jedes 3. Schülers ergibt

1							8					13
---	--	--	--	--	--	--	---	--	--	--	--	----

Elimination jedes 3. Schülers ergibt

							8					13
--	--	--	--	--	--	--	---	--	--	--	--	----

Elimination jedes 3. Schülers ergibt

												13
--	--	--	--	--	--	--	--	--	--	--	--	----

Nacheinander werden also eliminiert:

3, 6, 9, 12, 2, 7, 11, 4, 10, 5, 1, 8

Übrig bleibt die 13.

Also hat Schüler 13 gewonnen.

Aufgaben:

Schreiben Sie ein Programm, das von n Schülern so lange jeweils den k-ten Schüler eliminiert, bis nur noch ein (bzw. eine konstante Anzahl, die nicht mehr verringert werden kann) Schüler übrig bleibt.

58)

Der Tennisverein TSV Ballingen lädt zu einem Turnier mit $n = 16$ Teilnehmern ein, die jeweils im Doppel gegeneinander antreten.

Geben Sie alle mögliche Doppelpaarungen an.

59)

Erstellen Sie einen Taschenrechner, der ganze Zahlen addieren bzw, subtrahieren kann, die in 8 Byte passen.

60)

a) In einem Feld vorgegebener Länge sollen zufällige Primzahlen abgespeichert werden. (Erzeugen Sie dazu jeweils eine Zufallszahl. Ist diese eine Primzahl, wird sie abgespeichert, sonst nicht).

b) Zeigen Sie für jede dieser Primzahlen Folgendes:

24 teilt das um 1 verminderte eines jeden Quadrats einer Primzahl p oder formaler:

24 teilt $(p^2 - 1)$ bzw. $(p^2 - 1) / 24$ ist eine ganze Zahl.

61) Expandierendes System ("Urknall-Expansion")

a) Nach dem "Urknall" gibt es die 2 Atome (Bausteine) 3 und 5

3, 5

Die folgende Regel R gibt an, wie aus einer Menge von Bausteinen eine neue Menge entsteht, die die alte Menge enthält:

Addiere jeweils 2 Atome (die ungleich sein müssen, also nicht 2 gleiche Atome) aus der alten Menge und füge sie später der alten Menge hinzu (falls sie dort noch nicht vorkommt).

Beispiel:

3,5

Erweiterung1: 8 ergibt:

3,5, 8

Erweiterung2: 11, 13 ergibt:

3,5, 8, 11, 13

Erweiterung3: 14, 16, 19, 24, 18, 21 ergibt:

3,5, 8, 11, 13, 14, 16, 19, 24, 18, 21

...

Erstellen Sie ein Programm, das eine bestimmte, vorgegebene Anzahl Erweiterungen macht, das dazugehörige Feld erzeugt und auf dem Bildschirm ausgibt.

b)

Man kann andere, selbst erzeugte Zahlenmengen basteln:

V1) Es können andere Anfangszahlen (und eine andere Anzahl) benutzt werden.

V2) Es können andere Vorschriften benutzt werden, wie z.B:

$(x, y) \rightarrow x + y$, falls $x \neq y$

$(x, y) \rightarrow |x - y|$, falls $x \neq y$

$(x, y) \rightarrow x + y$, falls $x \neq y$ und nicht x und y beide gerade Zahlen

Man könnte sich folgende Fragen überlegen:

F1) Gibt es eine Zahl z , aber der alle darauffolgenden Zahlen $z+1, z+2, z+3, \dots$ (ohne Lücke) in der Menge der wunderbaren Zahlen vorkommen ?

F2) Kann man die Anfangszahlen so vorbelegen (und die Vorschrift so gestalten) , daß dies gelingt ?
F3) usw.

Noch in Bearbeitung:

x) Roulette

x) Funktionen wie strncpy, getch, usw. erklären.

a) Wie groß ist der Mittelwert aller Würfe eines bestimmten Würfels ?

b) Wie groß ist der Mittelwert aller Würfe, auf alle Würfel bezogen ?

c) Wie oft kam eine bestimmte Augenzahl in allen Würfeln eines Würfels vor (absolut und relativ).

d) Wie lang war die maximale Serie von Würfeln, in der eine bestimmte Augenzahl (z.B. 6) nicht mehr bei einem Würfel (z.B. Würfel 2) vorkam und bei welchem Wurf begann (und endete) sie ?

e) Wie lang war die maximale Serie von Würfeln, in der eine bestimmte Augenzahl (z.B. 6) nicht mehr bei allen Würfeln - auf insgesamt alle Würfel bezogen - vorkam und bei welchem Wurf begann (und endete) sie ?

f) Wie oft kam eine bestimmte Augenzahl in insgesamt allen Würfeln aller Würfel vor (absolut und relativ).

g) Wie oft kam eine bestimmte Augenzahl in einem bestimmten Wurf (z.B. dem Wurf 72) - auf alle 3 Würfel bezogen - vor ?

h) Wie oft kamen 3 verschiedene Augenzahlen in einem bestimmten Wurf (z.B. dem Wurf 72) - auf alle 3 Würfel bezogen - vor ?

x) In einem 3 x 3 Feld darf ein Spieler beliebig Steine setzen.

Wer zuerst 3 in einer Linie hat, hat gewonnen.

x) Simulation einer kapitalistischen Ökonomie

Auf einer Insel gibt es m Unternehmer. Jeder davon stellt Investitionsgüter (z.B. Maschinen, mit denen man Konsumgüter herstellt) her, indem er zuerst von einem anderen Unternehmer Investitionsgüter zu B Geldeinheiten (kurz GE) kauft, davon Investitionsgüter zu $2 \cdot B$ GE (allgemein $k \cdot B$ GE) herstellt und diese an die anderen (einen oder mehrere) Unternehmer verkauft. Die anderen Unternehmer verhalten sich analog.

Beispiel (mit 3 Unternehmern U_1, U_2, U_3):

Ausgangssituation:

U_1 hat Investitionsgüter im Wert von 2GE. Diese verkauft er an U_2 und U_3 .

Kontostand 0 (Ausgangssituation):

U_1	U_2	U_3
2GE	-1GE	-1GE

Vorgang:

U_2 und U_3 produzieren daraus jeweils Investitionsgüter vom doppelten Wert und verkaufen diese an U_1

Kontostand 1:

U_1	U_2	U_3
-2GE	1GE	1GE

Vorgang:

U_1 produziert daraus Investitionsgüter vom doppelten Wert und verkaufen diese nur an U_2

Kontostand 2:

U1	U2	U3
2GE	-3GE	1GE

Simulieren Sie diese Vorgänge, indem z.B. der Verkauf eines Unternehmers an die anderen Unternehmer durch den Zufall modelliert wird.

These:

Wenn die Anzahl n der Simulationsschritte wächst, wird es immer einen Unternehmerr geben, dessen Kontostand immer mehr abnimmt (dieser Unternehmer hängt von der Anzahl der Simulationen und der wirtschaftlichen Vorgänge ab).

Nehmen Sie zu diese These Stellung, indem Sie eine geeignete Simulation implementieren!

x)

siehe Aufgaben zu Dateien:

Diese Aufgaben übertragen auf Felder !!

