

Inhaltsverzeichnis

| | |
|---|----|
| 1. PROG-Array..... | 1 |
| 1.1. Dim1: int-Array..... | 2 |
| 1.1.1. RandomSucheMinimumHatDoppelte..... | 2 |
| 1.1.2. Rotate..... | 2 |
| 1.1.3. Palindrom..... | 2 |
| 1.1.4. Lotto..... | 3 |
| 1.1.5. MischenNachbar..... | 3 |
| 1.1.6. Permutation..... | 3 |
| 1.1.7. Geburtstag..... | 3 |
| 1.1.8. BubbleSort..... | 4 |
| 1.1.9. MergeSort..... | 4 |
| 1.1.10. Haeufigkeiten..... | 4 |
| 1.1.11. MittelwertMedianStandardabweichung..... | 5 |
| 1.1.12. Tonfilter..... | 5 |
| 1.2. Dim1: char Array..... | 6 |
| 1.2.1. Caesar..... | 6 |
| 1.2.2. Anagramm..... | 7 |
| 1.2.3. Ersetze..... | 7 |
| 1.2.4. Halt..... | 7 |
| 1.2.5. GetPasswd..... | 7 |
| 1.2.6. IndexOf..... | 8 |
| 1.2.7. HamAbstand..... | 8 |
| 1.2.8. HamAehnlich..... | 8 |
| 1.2.9. isbn..... | 8 |
| 1.2.10. Zwei2Eins..... | 9 |
| 1.2.11. Lesbar..... | 9 |
| 1.3. Dim2: Matrix..... | 9 |
| 1.3.1. MatrixAB..... | 9 |
| 1.3.2. Maeander..... | 10 |
| 1.3.3. AddTable..... | 10 |
| 1.3.4. MulTable..... | 10 |
| 1.3.5. TransMatrix..... | 10 |
| 1.3.6. MagischesQuadrat..... | 10 |
| 1.3.7. AddMatrix..... | 11 |
| 1.3.8. MulMatrix..... | 11 |
| 1.3.9. ShortestPath.java..... | 11 |

1. PROG-Array

☒ Dokumentieren Sie ihr Programm ausreichend!

```
/** Hausübung: Arrays
 * ??? Hier kommt die Kurzbeschreibung her ?????
 * @author ?????, inkl. evtl. Verweis auf Hilfe durch KollegInnen
 * @date ?????
 * @file ?????
 * Meine Zeitschätzung: ?????
 * Tatsächlich gebrauchte Zeit: ???
 */
```

Hinweis: Zufallszahlen in C

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

```
....  
int arr[10];  
int i;  
  
srand(time(NULL)); //für immer neue Zufallszahlen  
for (i=0; i < 10; i++){  
    arr[i]= rand()%45 + 1; // 1 .. 45  
}
```

Hinweis: Zufallszahlen in Java

```
int[] arr= new int[10];  
Random zufall= new Random();  
for (int i=0; i < 10; i++){  
    arr[i]= zufall.nextInt(100); // 0..99  
}
```

1.1. Dim1: int-Array

1.1.1. RandomSucheMinimumHatDoppelte

1. Schreiben Sie Zufallszahlen (Bereich 0..99) in ein 10-elementiges Array.
2. Suchen Sie im Array nach der Zahl 50. Wenn sie gefunden wurde, soll der Index (also die Position im Array) ausgegeben werden. Ansonsten wird ausgegeben, dass die Zahl 50 nicht im Array ist.
3. Suchen Sie den Index der kleinsten Zahl im Array und geben Sie diesen aus.
4. Geben Sie auch aus, ob im Array Werte doppelt vorkommen.

1.1.2. Rotate

1. Definiere ein int-Array namens a mit 10 Elementen.
2. Schreib mit Hilfe einer for-Schleife die werte 0,1,2,... ins Array
3. Gib das Array aus.
4. Lies von der Tastatur einen int-Wert ein in die Variable index
5. Rotiere alle Arrayelemente um index-Stellen
6. Gib das Array aus

Beispiel: Für index wurde 1 eingegeben

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

1, 2, 3, 4, 5, 6, 7, 8, 9, 0

Hinweis:

Rotieren bedeutet:

1. Erstes Arrayelement in eine Hilfsvariable speichern
2. Alle anderen Arrayelemente im Array um eine Stelle nach vorn verschieben
3. zwischengespeicherten Wert hinten ins Array schreiben

1.1.3. Palindrom

1. Lies Zahlen in ein Array ein und
2. bestimme ob es sich um ein Palindrom (vorwärts gelesen ist gleich rückwärts gelesen.) handelt.

Beispiel:

123321 oder 101 oder 11011 sind Palindrome

1.1.4. Lotto

Das Programm

1. definiert das Array `lotto_mein_tipp` mit genau 6 Elementen vom Typ `int`
2. schreibt **6 garantiert verschiedene Zahlen** im Bereich 1 bis 45 (inkl.) in das Array.
3. gibt das Array aus.
4. definiert das Array `lotto_ziehung` mit genau 6 Elementen vom Typ `int`
5. schreibt **6 garantiert verschiedene Zahlen** im Bereich 1 bis 45 (inkl.) in das Array.
6. gibt das Array aus und
7. gibt Ihren eventuellen Gewinn aus. (0er,1er,2er,...)

1.1.5. MischenNachbar

1. Definieren Sie ein `int`-Array mit genau 10 Elementen
2. schreiben Sie mit einer `for`-Schleife in das Arrayelement an der Stelle 0 den Wert 0, in das Arrayelement an der Stelle 1 den Wert 1, usw.
3. geben Sie das Array aus.
4. mischen Sie nun die Arrayelemente auf folgende Weise:
 1. 100 mal
ermittle für die Variable `iStelle` eine Zufallszahl im Intervall `[0,ANZ]`.

ermittle für die Variable `jStelle` eine Zufallszahl im Intervall `[0,ANZ]`.

tausche die beiden Arrayelemente an der Stelle `iStelle` und `jStelle`.
5. gib das Array aus
6. Ermitteln Sie, ob im Array Nachbarn (Bsp: 5→ 6 oder 8→9, ... sind Nachbarn) vorkommen.

1.1.6. Permutation

Ein Array `int [n]` enthält eine »Permutation«, wenn alle Zahlen 1, ... , n genau einmal als Elemente vorkommen.

Aufgabe: Permutation

Schreiben Sie ein Programm (`permutation`), das

☒ ein Array namens `permutation` mit genau 4 Elementen definiert

☒ lesen Sie von der Tastatur 4 Elemente ins Array ein.

Achtung die Werte dürfen nur zwischen 0 bis 4 sein. D.h. Ihr Programm darf keine anderen Werte ins Array einlesen.

☒ Überprüfen Sie, ob die eingegebenen Werte im Array eine Permutation ergeben.

Eine 0 spielt dabei die Rolle eines Jokers, der beliebige Werte 1, ... , n annehmen kann.

☒ Geben Sie nach der Überprüfung das Array und das Ergebnis ihrer Überprüfung aus. (siehe Beispiel unten)

Beispiele:

(2,1,4,3) ist eine Permutation der Zahlen 1, ... ,4.

(1,2,2,3) ist keine Permutation.

(2,0,0,3) kann zur Permutation ergänzt werden.

(2,0,0,2) kann nicht zu einer Permutation ergänzt werden.

1.1.7. Geburtstag

Schreiben Sie ein Programm, das den Geburtstag (int tag, int monat, int jahr) einliest und den entsprechenden Wochentag ausgibt. Achten Sie auch darauf, dass nur gültige Werte eingegeben werden können.

☒ input:

int tag, int monat, int jahr (geburtstag)

☒ output:

der dd.mm.jjjj ist ein (montag, dienstag, ...)

Hinweis: Formel zur Berechnung des Wochentages

$w = [a + \text{int}((a/4) - \text{int}(a/100) + \text{int}(a/400) + d) \bmod 7;$

a jahreszahl des vorjahres

d anzahl der tage im gefragten jahr (inkl. gewuenschter tag)

w ist der Wochentag mit

w = 0 so w = 1 mo w = 2 di w = 3 mi w = 4 do w = 5 fr w = 6 sa

Beispiel: Auf welchen Wochentag fiel der 1.03.1984

a=1983

d= 31+29+1=61 (1984 ist ein Schaltjahr)

Hinweise zum Schaltjahr:

1. durch 4 teilbar

2. bei vollem Jahrhundert, das durch 400 teilbar ist,

handelt es sich um ein schaltjahr, sonst nicht

bsp: 1900 kein schaltjahr aber 2000 ein schaltjahr

if ((jahr%4 ==0) && (jahr%100 !=0) || (jahr%400 ==0))

// jahr ist ein schaltjahr

else

//jahr ist kein schaltjahr

Hinweis: PAP - Programmablaufplan

1. tag (1-31), monat (1-12), jahr einlesen

2. vorjahr ermitteln

3. anzahl_der_tage ermitteln

4. formel berechnen

5. ausgabe

Hinweis: Zur Berechnung der Tage im Jahr verwenden Sie

int tage_im_jahr[]={0,31,28,31,30,31,30,31,31,30,31,30,31}; // hier in C

1.1.8. BubbleSort

1. Schreiben Sie in ein int-Array 10 Zufallszahlen (0..99).
2. Sortieren Sie dieses Array nach dem BubbleSort-Algorithmus (siehe: <https://de.wikipedia.org/wiki/Bubblesort>)
3. Geben Sie das sortierte Array zur Kontrolle aus.

1.1.9. MergeSort

Schreiben Sie ein Programm, dass 2 schon sortierte Arrays in ein Array zusammenmischt, sodass das neue Array auch sortiert ist.

1.1.10. Häufigkeiten

1. Schreiben Sie in ein int-Array 10 Zufallszahlen (0..99)
2. Berechnen Sie die absoluten Häufigkeiten (=Anzahl) der jeweiligen Zahlen im Array.
3. Berechnen Sie die relative Häufigkeiten und
4. Geben Sie die obigen Arrays aus.

Hinweis: https://de.wikipedia.org/wiki/Absolute_H%C3%A4ufigkeit

1.1.11. MittelwertMedianStandardabweichung

1. Schreiben Sie in ein int-Array 10 Zufallszahlen (0..99)
2. Berechnen Sie Mittelwert, Median und die Standardabweichung
3. Geben Sie das Array und die Kennzahlen aus.

Hinweis: Standardabweichung

$$s_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}.$$

1.1.12. Tonfilter

Ein Tonsignal wird manchmal als Liste von int-Werten gespeichert. Die Werte repräsentieren die Intensität des Signals in aufeinanderfolgenden Zeitintervallen. Natürlich wird in einem Programm das Signal durch ein Array repräsentiert.

Oft ist in dem Signal ein kleiner Anteil von Störgeräuschen enthalten. Störgeräusche sind üblicherweise kleine, momentane Änderungen der Tonhöhe. Ein Beispiel ist das "Geräusch", das zusätzlich zum Ton eines AM Radios zu hören ist.

Das Glätten des Tons entfernt das Störgeräusch und verbessert die Wahrnehmung der Tonqualität. Diese Aufgabe besteht darin, die Werte eines Integerarrays zu glätten.

Angenommen, dass die ursprünglichen Werte in dem Array "signal" sind. Berechnen Sie das geglättete Array, in dem Sie folgendes tun: Jeder Wert `geglatttet[N]` ist der Durchschnitt von drei Werten: `signal[N-1]`, `signal[N]` und `signal[N+1]`.

Berechnen Sie für das erste Element von `geglatttet` den Durchschnitt der ersten zwei Elemente von `signal`. Berechnen Sie für das letzte Element von `geglatttet` den Durchschnitt der letzten zwei Elemente von `signal`.

Verwenden Sie dafür Integerarithmetik, so dass die Werte in `geglatttet` Integer sind.

```
#define ANZ 12
int main ( )
{
    intj;
    int signal[ANZ] = {5, 5, 4, 5, 6, 6, 7, 6, 5, 4, 1, 4};
    int geglaettet[ANZ];

    // berechnen Sie den geglätteten Wert für jeden Slot
    // von Array geglaettet
    geglaettet[0] = ???????
    geglaettet[ ANZ-1 ] = ??????
    for ( ???? )
    {
        ?????????
    }
}
```

```
// Geben Sie den Input aus
for ( j = 0; j < ANZ; j++)
{

}

// Geben Sie das Ergebnis aus
for ( int j = 0; j < ANZ; j++)
{

}

}
}
```

Denken Sie daran, wenn Sie die Ergebnisse interpretieren, dass Integerdivision den Rest verwirft. Sie berechnet keinen gerundeten Wert. Hier ist, was das Programm ausgibt.

```
./tonfilter.exe
signal: 1 5 4 5 7 6 8 6 5 4 5 4
gegläutet: 3 3 4 5 6 7 6 6 5 4 4 4
```

1.2. Dim1: char Array

Hinweis: Java-Array-Strings:

```
String s1= "Hallo";
char[] sArr= s1.toCharArray();
char ch1= s1.charAt(0); // H
char ch2= sArr[0];
sArr[0]= 'X';
String s2= new String(sArr);
String s3= String.valueOf(sArr);
boolean istGleich= s1.equals(s2); // Strings vergleichen: liefert true oder false
```

1.2.1. Caesar

Cäsar-Chiffre: Zur Kodierung seiner Nachrichten soll Cäsar folgendes Verfahren angewandt haben: Statt des Buchstabens, den er meinte, schrieb er den Buchstaben auf, der im Alphabet k Positionen rechts vom gemeinten Buchstaben ist. Wenn man bei 'z' ankommt, muss man bei 'a' weiter zählen.

Beispiel: Mit $k=4$ wird aus "hello world" der verschlüsselte Text "lipps asvph".

Schreiben Sie ein Programm (Text einlesen, Text verschlüsseln), das **nur** die Buchstaben mit dem Cäsar-Chiffre verschlüsselt, aber alle anderen Zeichen unverschlüsselt lässt. Aus "Hello, world!" soll also "Lipps, asvph!" werden.

Anmerkung: Wenn man zB. zum Zeichen Z kommt muss man beim Zeichen A weiter zählen. (analog z -> a)

```
ch=ch+key;
if(ch>'Z')
```

```
ch= ch-26;  
  
bzw.  
ch=ch+key;  
if(ch>'Z')  
    ch= ch-26;
```

Anmerkung: Entschlüsselt wird mit $\text{key}=26-\text{key}$;

Beispiel:
Hello, world! (key=4)
Lipps, asvph! (entschlüsselt wird mit 22) (vgl: $26 - 4$)

Hinweis:
Als Datentyp verwenden Sie unsigned char. (in C)

1.2.2. Anagramm

Lesen Sie 2 Strings ein und bestimmen Sie, ob es sich um ein Anagramm handelt.

Hinweis: MEHL ist ein Anagramm zu HELM.

- AMPEL ist ein Anagramm zu LAMPE und PALME.
- VENUS IN BETON ist ein Anagramm zu SUBVENTIONEN.
- NAJA MILCHSOCKE ist ein Anagramm zu MICHAEL JACKSON

1.2.3. Ersetze

Gegeben sei folgender Programmcode (Hier in C):

```
int main()  
{    char s1[] = "ueb immer treu und redlichkeit";  
    char s2[] = "opqrstu";  
  
    // ??????????????????  
}
```

Es soll nun der Text s1 geändert werden, und zwar auf folgende Art:
Jedes Zeichen im Text s1, das im Text s2 vorkommt, soll durch '*' ersetzt werden.

Im obigen Beispiel würde s1 dann beinhalten:
eb imme **e* *nd *edlichkei*

Die im Programm angegebenen Inhalte von s1 und s2 sind nur Beispiele, es könnte sich auch um andere Texte handeln.

1.2.4. Halt

1. Lies einen Text ein und gibt diesen wieder aus, aber nur bis dass die Wortfolge halt ausgegeben wurde. Anmerkung die Buchstaben halt müssen nicht hintereinander im Text vorkommen

Beispiel:
ABHFTAZRLJQWTZBE

ABHFTAZRLJQWT

1.2.5. GetPasswd

Liest eine Länge für einen Passwort-Vorschlag ein und gib dann ein Passwort aus, das aus folgendem Zeichenvorrat Zeichen wählt: Gross, Kleinbuchstaben, Ziffern, Sonderzeichen „.-<>,:;-_#+*!\$%&/()=?“

1.2.6. IndexOf

Gegeben sei folgender Programmcode (Hier in C):

```
int main() {
    char s1[] = "ueb immer treu und redlichkeit";
    char s2[] = "immer";
    int index=-1;

    // ????????????????
}
```

Ermittle den index in s1, an dem s2 vorkommt.
Im obigen Beispiel würde index dann den Wert 4 haben.

Die im Programm angegebenen Inhalte von s1 und s2 sind nur Beispiele, es könnte sich auch um andere Texte handeln.

1.2.7. HamAbstand

Schreiben Sie ein Programm, das den sogenannten Hamming-Abstand zweier Strings berechnet. Der Hamming-Abstand der beiden Zeichenketten PUPPENKISTEN und SUPPENKASPER ergibt beispielsweise 4 und entspricht der Anzahl der 4 nicht übereinstimmenden Zeichen:

| | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|
| A: | P | U | P | P | E | N | K | I | S | T | E | N |
| | | | | | | | | | | | | |
| B: | S | U | P | P | E | N | K | A | S | P | E | R |

Die acht übereinstimmenden und untereinanderstehenden Zeichen der Sequenzen A und B sind mit einem '|' gekennzeichnet.

1.2.8. HamAehnlich

Um die Hamming-Ähnlichkeit A_H zweier Zeichenketten $A := a_1a_2\dots a_n$ und $B := b_1b_2\dots b_n$ auszudrücken, wird der Hamming- Abstand $D_H(A;B)$ in Relation zur Gesamtlänge n der beiden Zeichenketten gesetzt und dieser Wert von 1 subtrahiert:

$$A_H(A, B) := 1 - \frac{D_H(A, B)}{n}$$

Würden alle Zeichen übereinstimmen, so entspräche dies dem Wert 1. Käme es zu keiner Übereinstimmung so entspräche dies einer Hamming-Ähnlichkeit von 0. Für die Sequenzen A und B aus dem obigen Beispiel ergäbe die Berechnung der Hamming-Ähnlichkeit $1-4/12$, also das Ergebnis 0,66.

1.2.9. isbn

Zur Kennzeichnung von Waren verwendet man den sogenannten EAN-Code (Europäische Artikel-Nummerierung mit 13 Ziffern). Bei Büchern ist die ISBN-Nummer üblich. Bei der letzten Ziffer der Nummer handelt es sich um eine sogenannte Prüfziffer, sodass z.B. einfache Eingabefehler erkannt werden können. Die Prüfziffer berechnet sich aus den übrigen Ziffern.

ISBN-13

Zur Berechnung der Prüfziffer bei der ISBN-13 werden alle zwölf Ziffern der noch unvollständigen ISBN addiert, wobei die Ziffern mit gerader Position (also die 2., 4. usw.) dreifachen Wert erhalten.

Bsp: Eine 5 an 6. Stelle beispielsweise fließt als 15 in die Addition ein.

Von dem Ergebnis dieser Addition wird die letzte Stelle bestimmt, die dann von 10 subtrahiert wird.

Bsp: Also etwa $10 - 4 = 6$ bei einem Additionsergebnis von 124.

Dieses Endergebnis ist die Prüfziffer. Ist das Endergebnis indessen 10, ist die Prüfziffer 0.

Formel zur Berechnung der Prüfziffer:

$$z_{13} = 10 - \left(\sum_{i=1}^{n=12} z_i \cdot 3^{(i+1) \bmod 2} \right) \bmod 10$$

Das $(i+1) \bmod 2$ sorgt für die wechselnde Gewichtung von 1 und 3.

Erstreckt man die Summierung auch auf die Prüfziffer ($n = 13$), so erhält man bei einer fehlerfreien ISBN als Ergebnis 0.

Beispiel:

978-3-7657-2781-?

Lösung:

$$9 + 8 + 7 + 5 + 2 + 8 + 3 * (7 + 3 + 6 + 7 + 7 + 1) = 39 + 3 * 31 = 39 + 93 = 132$$

$$132 \bmod 10 = 2$$

$$10 - 2 \bmod 10 = 8 \quad \text{d.h. Die Prüfziffer ist 8}$$

1.2.10. Zwei2Eins

In diesem Programm soll aus 2 eingegeben char-Arrays/string ein Array/string gemacht werden. Das neue Array/String soll aber die doppelten Buchstaben nicht wiederholen.

Also z.B

[A]=A B C D E F

[B]=D E F G H I

das neue Array soll so aussehen

[C]=A B C D E F G H I

1.2.11. Lesbar

Lesen Sie wortweise ein und tauschen Sie die Buchstaben des Wortes beliebig oft. Aber lassen Sie den 1. und letzten Buchstaben unvertauscht.

Geben Sie die Wörter aus. Kann man den so erhaltenen Text noch lesen?

1.3. Dim2: Matrix

1.3.1. MatrixAB

Lesen Sie eine Zahl ein (darf nur zwischen 3 und 10) sein:

Erstellen Sie eine 2-dim Matrix und schreiben Sie A und B auf folg. Weise ein und geben Sie die Matrix aus:

Beispiel: (4 wurde eingegeben)

ABABAB

BABABA

ABABAB

BABABA

1.3.2. Maeander

Erzeugen Sie eine zweidimensionale (int) Matrix, mit je n Spalten und Zeilen und befüllen Sie die Matrix zeilenweise von links nach rechts, bzw. von rechts nach links alternierend (mäandertförmig) aufsteigend mit den natürlichen Zahlen. Geben Sie es mit den jeweiligen Quersummen der Zeilen wie abgebildet aus. Definieren Sie n als leicht änderbare Konstante, sodass das Verfahren mit beliebigen $n > 2$ funktioniert.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | sum |
|---|----|----|----|----|----|----|----|----|-----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 36 |
| 2 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 100 |
| 3 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 164 |
| 4 | 32 | | | | | | | | |
| 5 | 33 | | | | | | | | |
| 6 | 48 | | | | | | | | |
| 7 | 49 | | | | | | | | |
| 8 | 64 | | | | | | | | |

1.3.3. AddTable

Lesen Sie eine Dimension ein (zw. 3 und 10)

Erstellen Sie eine sog. Additions-Tabelle nach dem Beispiel unten.

| | 0 | 1 | 2 | ... |
|----|-----|-----|----|-----|
| 0 | 0 | 1 | 2 | ... |
| 1 | 1 | 2 | 3 | ... |
| 2 | 2 | 3 | 4 | ... |
| .. | ... | ... | .. | ... |

1.3.4. MulTable

wie oben aber eine Multiplikations-Tabelle.

1.3.5. TransMatrix

Schreiben Sie ein Programm, dass eine quadratische Matrix transponiert. Beim Transponieren, wird die Matrix an ihrer Hauptdiagonale gespiegelt. Das Element a , dass in der i -ten Zeile und j -ten Spalte steht, steht in der transponierten Matrix in der j -ten Zeile und i -ten Spalte.

1.3.6. MagischesQuadrat

Ein magisches Quadrat ist eine quadratische Anordnung mit n Zeilen und n Spalten von Zahlen derart, dass die Summe der Zahlen in einer beliebigen Zeile, Spalte, oder in der Hauptdiagonale alle gleich sind.

| | | |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

Wenn n ungerade ist, kann man ein magisches Quadrat folgendermaßen erstellen. Im Feld $Q[i][j]$ ($i=1..n, j=1..n$) wird eingetragen:

1. Setzen Sie $k = j - i + (n - 1)/2$ und $m = 2*j - i$.
2. Wird $k \geq n$, ersetzen Sie k durch $k - n$ und fahren Sie fort bei Schritt 4.
3. Wird $k < 0$, ersetzen Sie k durch $k + n$.
4. Wird $m > n$, so ersetzen Sie m durch $m - n$ und fahren Sie fort bei Schritt 6.
5. Wird $m \leq 0$, ersetzen Sie m durch $m + n$.
6. Ergebnis: $Q[i][j] = K*n + m$.

Erstellen Sie ein Programm, das mit diesen Regeln bei Benutzereingabe von ungeradem n ein magisches Quadrat erzeugt, das in einem Array gespeichert und dann am Bildschirm schön formatiert ausgegeben wird.

1.3.7. AddMatrix

Lies int-Werte für 2 Matrizen ein (3*3). Kann auch durch Zufallszahlen geschehen.

Gib die Matrizen aus und

Berechne die Multiplikation ($C = A+B$) ($c_{ij} = a_{ij} + b_{ij}$)

Gib die Matrix C aus.

1.3.8. MulMatrix

Lies int-Werte für 2 Matrizen ein (3x3). Kann auch durch Zufallszahlen geschehen.

Gib die Matrizen aus und

Berechne die Multiplikation ($C = A*B$) ($c_{ij} = \text{summe}(a_{ik} * b_{kj})$)

Gib die Matrix C aus.

1.3.9. ShortestPath.java

todo

