

Aufgabenblatt 4

Letzter Abgabetermin: Finden Sie im Kalender im RELAX

Achten Sie darauf, dass die Endung Ihrer Quellcode Datei .c lautet und **nicht** .cpp

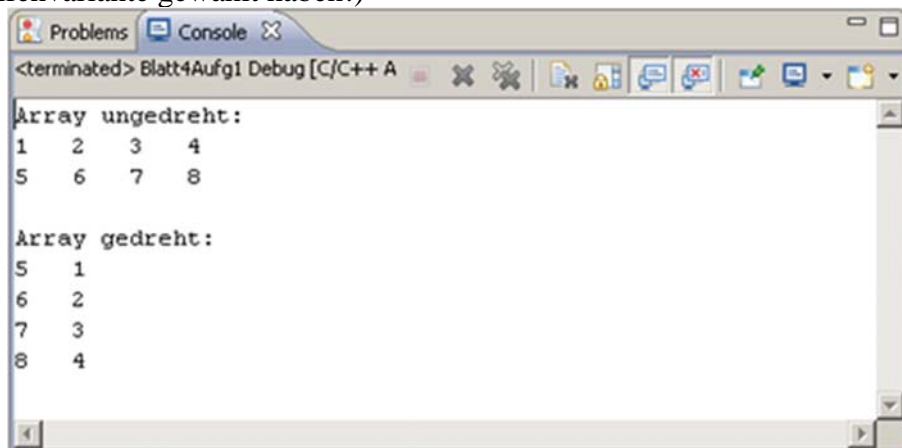
Verwenden Sie für Variablen **sinnvolle** Bezeichnungen!

Schwerpunkt des Aufgabenblattes:

- Arrays
- Strukturen
- Pointer

1. [5 Punkt]

Schreiben Sie ein Programm, das ein zweidimensionales *int*-Array anlegt, es initialisiert und das Array dann um 90° dreht. (Verwenden Sie sinnvolle Schleifen! Erklären Sie warum Sie Ihre Schleifenvariante gewählt haben!)



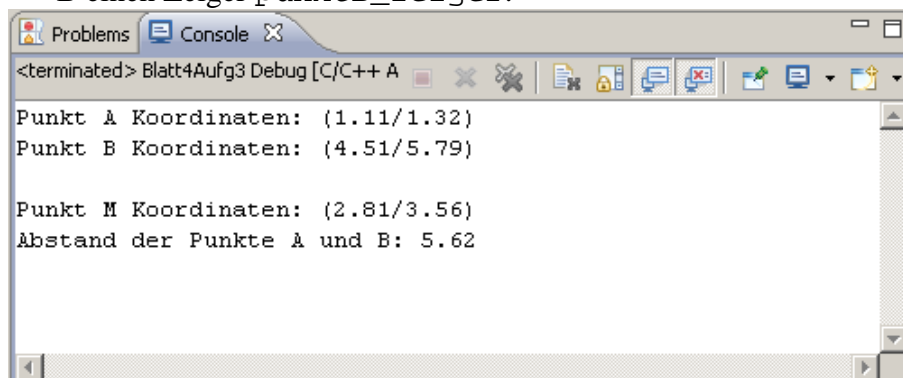
```
<terminated> Blatt4Aufg1 Debug [C/C++ A
Array ungedreht:
1 2 3 4
5 6 7 8

Array gedreht:
5 1
6 2
7 3
8 4
```

2. [5 Punkt]

Schreiben Sie ein Programm das den Mittelpunkt, sowie die Entfernung zweier Punkte berechnet, speichert und ausgibt.

- Nutzen Sie zum Speichern der Punkte eine Struktur Punkt, die zwei Komponenten x und y vom Typ *double* enthält.
- Nutzen Sie für den Punkt A eine Variable punktA vom Typ der Struktur und für Punkt B einen Zeiger punktB_zeiger.



```
<terminated> Blatt4Aufg3 Debug [C/C++ A
Punkt A Koordinaten: (1.11/1.32)
Punkt B Koordinaten: (4.51/5.79)

Punkt M Koordinaten: (2.81/3.56)
Abstand der Punkte A und B: 5.62
```

3. [10 Punkte]

Schreiben Sie auf Grundlage des Telefonverzeichnisses von Blatt 3 Aufgabe 2 ein Programm, das zählt wie oft eine bestimmte Option gewählt wurde. Zählen Sie auch falsche Eingaben mit.

Nutzen Sie dazu eine Struktur die jeweils die Komponenten eines Telefonbucheintrags speichern kann. Dazu werden folgende Komponenten benötigt:

- `String`, für den Text in Langform (*char*-Array mit abschließendem `,\0'`)
- Zeichen, für den zugehörigen Buchstaben
- `Anzahl` (positive ganze Zahl), wie oft diese Option gewählt wurde

Nach Beenden des Programms soll eine Tabelle ausgegeben werden welche Option wie oft gewählt wurde. Sortieren Sie die Tabelle nach der Häufigkeit der Aufrufe.

Hinweis: Speichern Sie die Einträge in einem Array vom Typ der Struktur damit Sie einen einfachen Sortieralgorithmus anwenden können (Beispielsweise Bubble-Sort, Google/Wikipedia hilft).

Anzahl Auswahl	Buchstabe Eintrag	Langtext Eintrag
5		Falsche Eingabe
3	E	Neuen Eintrag einfüegen
1	B	Programm beenden
0	L	Eintrag loeschen
0	S	Telefonnummer suchen
0	A	Alle Eintraege anzeigen

4. [10 Punkt]

Gegeben sei das Telefonverzeichnis vom Blatt 3 Aufgabe 2 und eine Struktur Eintrag mit den nachfolgenden Komponenten

```
struct Eintrag *naechster
int nr
char name [ 20 ]
```

Schreiben Sie ein Programm, welches zur Laufzeit in der Reihenfolge der Eingabe dynamisch die nachfolgenden Einträge erzeugt.

Hinweis: Dies wird auch als einfach verkettete Liste bezeichnet.

naechster	nr	name
Zeiger auf Struktur mit Nr. 34	12	Meier
Zeiger auf Struktur mit Nr. 56	34	Mueller
NULL	56	Schulze

Das Programm soll nun die Aufgaben des Telefonverzeichnisses wie folgt lösen:

a) Fügen Sie bei Auswahl „E“ (Neuen Eintrag einfüegen) am Ende der Liste einen neuen Eintrag ein mit einer neuen Nummer und Ihrem Namen. Geben Sie anschließend alle Einträge der Reihenfolge nach aus.

[Optional] b) Geben Sie bei Auswahl „S“ (Telefonnummer suchen) eine Telefonnummer ein und prüfen Sie, ob diese in der Liste enthalten ist.

Hinweis: Um die Funktionalität prüfen zu können, wäre es natürlich sinnvoll die Auswahl „A“ (Alle Eintraege anzeigen) ebenfalls zu implementieren.