

Zeiger auf Funktionen

In dieser Aufgabe ist ein Feld zu sortieren. C stellt zum Sortieren in der Standardbibliothek *stdlib.h* eine universelle Sortieroutine `qsort()` zur Verfügung. Der Prototyp dieser Funktion lautet

```
void qsort(void *feld, long feldElementeAnzahl, long feldElementGroesse,
           int (*compare)(void* a, void* b));
// feld: Das zu sortierende Feld
// feldElementeAnzahl: Anzahl der Elemente im Feld
// feldElementGroesse: Groesse eines einzelnen Feldelements (in Bytes)
// int (*compare)(void*, void*): Zeiger auf Vergleichsfunktion, s.u.
```

Der letzte Parameter ist (ein Zeiger auf) eine Funktion, die zwei Feldobjekte `a` und `b` miteinander vergleicht. Rückgabewert dieser Vergleichsfunktion ist eine ganze Zahl `z` vom Typ `int`. Dabei gilt

```
z = 0 falls a = b
z = -1 falls a < b
z = 1 falls a > b
```

Um nun die Elemente eines Feldes nach einem vorgegebenen Kriterium zu sortieren, muß lediglich eine Vergleichsfunktion dieser Art programmiert und als letzter Parameter an `qsort` übergeben werden.

Es sind die 10 Zahlenpaare (x, y) einer Meßreihe zu sortieren. Dabei sind x und y beliebige **ganze** Zahlen. In welcher Reihenfolge zwei beliebige Zahlenpaare (x_1, y_1) und (x_2, y_2) dieser Meßreihe stehen, entscheiden die nachfolgend aufgeführten Kriterien.

- (x_1, y_1) steht vor (x_2, y_2) , falls $x_1^2 + y_1^2 < x_2^2 + y_2^2$
- Sind die Betragsquadrate gleich, so steht (x_1, y_1) vor (x_2, y_2) , falls $|x_1 - y_1| > |x_2 - y_2|$ ist.
- Andernfalls sind die beiden Wertepaare gleich(wertig)

Ergänzen Sie das nachstehende Programm um die fehlende Vergleichsfunktion `comparePaar`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define FELD_LAENGE 10
typedef struct {
    long x;
    long y;
} TZahlenpaar;

int main(void){
    // Beispieltabelle mit zehn Zahlenpaaren!
    TZahlenpaar messReihe[]={
        {-11,3},{2,6},{-1,1},{-1,-3},{-1,-4},
        {4,-7},{7,5},{-2,-9},{3,-1},{0,0}
    };
    qsort(messReihe, FELD_LAENGE,
          sizeof(TZahlenpaar), comparePaar);
    // Ausgabe der Tabelle
    // EBENFALLS von Ihnen zu erstellen!
    return 0;
}
// Vergleichsfunktion
// (von Ihnen zu erstellen!!!)
...
...
}
```

Bearbeitungshinweise

- Umwandlung von `void *a` in `TZahlenpaar *t`

```
t = (TZahlenpaar*) a;
```

- Zugriff auf die x -Komponente von `TZahlenpaar* t`

```
t->x
```