1 2D Arrays

Implementieren Sie eine Funktion, die einen 2D Array als Matrix auf der Konsole ausgeben kann. Diese Funktion soll mit folgendem Aufruf nutzbar sein:

```c
#include <stdio.h>

void print2dArray(size_t height, size_t width, /*TODO*/) {
    //TODO
}

int main() {
    const size_t width = 4, height = 3;
    double data[height][width];
    for(size_t y = 0; y < height; y++) {
        for(size_t x = 0; x < width; x++) {
            data[y][x] = 0;
            if(y == 1 && x == 1) data[y][x] = 4;
            if(y == 1 && x == 2) data[y][x] = 2;
        }
    }

    print2dArray(height, width, data);
}
```

Erwartete Ausgabe:

```
0.000000 0.000000 0.000000 0.000000
0.000000 4.000000 2.000000 0.000000
0.000000 0.000000 0.000000 0.000000
```

2 Callbacks

Bewertete Aufgabe!
Implementieren Sie eine Funktion, der Sie eine Callback-Funktion übergeben können. Die Callback-Funktion soll die Signatur `double foo(double x, double y)` haben. Außer dem Callback-Parameter soll die Funktion ein 2D Array übergeben bekommen. Ziel ist es, die Rückgabewerte der Callback-Funktion in den übergebenen Array zu schreiben. Die Funktion soll mit folgendem Code aufrufbar sein:

```c
#include <assert.h>
#include <stdlib.h>

double foo(double x, double y) {
    return x*x + y;
}

```

```
8  //Fills a 2D matrix with the results of the given callback function.
9  //The coordinates that are passed to the callback
10 //range from (y_min, x_min) to (y_max, x_max), inclusive.
11 //
12 //I.e. the call
13 //
14 //    fillMatrixWithResults(3, 0.0, 1.0,
15 //                          4, -1.0, 5.0,
16 //                          matrix, &callback)
17 //
18 //will call the callback with the coordinates
19 //
20 //    (0.0, -1.0) (0.0, 1.0) (0.0, 3.0) (0.0, 5.0)
21 //    (0.5, -1.0) (0.5, 1.0) (0.5, 3.0) (0.5, 5.0)
22 //    (1.0, -1.0) (1.0, 1.0) (1.0, 3.0) (1.0, 5.0)
23 void fillMatrixWithResults(int height, double y_min, double y_max,
24                            int width, double x_min, double x_max,
25                            /*TODO*/, /*TODO*/) {
26     //TODO
27 }
28
29 int main() {
30     const size_t height = 2, width = 3;
31     double data[height][width];
32     const double x_min = -1, x_max = 1;
33     const double y_min = 1, y_max = 3;
34
35     fillMatrixWithResults(height, y_min, y_max,
36                           width, x_min, x_max,
37                           data, &foo);
38
39     //may be used, is not required for grading
40     //print2dArray(height, width, data);
41
42     assert(data[0][0] == 2);
43     assert(data[0][1] == 1);
44     assert(data[0][2] == 2);
45     assert(data[1][0] == 4);
46     assert(data[1][1] == 3);
47     assert(data[1][2] == 4);
48 }
```

Achten Sie darauf, exakt das Verhalten zu implementieren, das in dem Funktionskommentar vorgegeben ist.

3 Integration

Schreiben Sie ein Programm, dass die beiden Funktionen aus den vorigen Aufgaben nutzt um die Werte folgender Funktionen auf die Konsole auszugeben:

- $f(x,y) = x \cdot y$ im Bereich $x,y \in [-1,1]$, Schrittweite $\frac{1}{2}$
- $g(x,y) = sinx \cdot siny$ im Bereich $x,y \in [0,\pi]$, Schrittweite $\frac{\pi}{4}$

- $h(x, y) = max(x, y)$ im Bereich $x, y \in [0, 10]$, Schrittweite 1

Hinweis: Um in C die Funktion `sin()` verwenden zu können, muss der Header `math.h` includiert werden und das Programm mit dem Flag `-lm` gelinkt werden werden. Dabei muss `-lm` nach den Objektdateien/`.c` Dateien angegeben werden. Eine Funktion `max()` gibt es nicht in der Standardbibliothek, die müsst Ihr also selber programmieren.

4 Abgabe

Abzugeben ist ein gemäß den bekannten Richtlinien erstelltes und benanntes Archiv. Das enthaltene und gewohnt benannte Verzeichnis soll folgenden Inhalt haben:

- Alle Quellen, aus denen Ihr Programm besteht (exercise07-2.c); gut dokumentiert(Kommentare im Code!)

Senden Sie Ihre Abgabe an cp-abgabe@wr.informatik.uni-hamburg.de .