Übung 2

Erarbeitet von: M.Eng. Michael Finsterbusch
Modulverantwortlicher: Prof. Dr. rer. nat. Matthias Krause
Stand: 3. Oktober 2017

Ziel der Übung ist das Kennenlernen und Vertiefen von:

- Schleifen
- Arrays
- einfache Berechnungen
- Verwenden von Funktionen

Abgabe:
- `http://praktomat.hft-leipzig.de` unter „Tutorial: DKMI/DAI-17 C-Progr.“
- sämtliche Abgabemodalitäten sind im Praktomat hinterlegt

Aufgaben

Gegeben ist die Header-Datei *uebung2.h*, in der Funktionsdeklarationen enthalten sind, sowie die Datei *uebung2.c*, in der die Funktionen implementiert werden. Kopieren Sie diese in Ihr Arbeitsverzeichnis. Die Funktionen sollen entsprechen den folgenden Vorgaben implementiert werden. Um die korrekte Funktionsweise Ihrer Implementation zu testen, verwenden Sie die Funktionen in der `main()`-Funktion, die in der Datei *main.c* implementiert werden soll. Diese kann zum Beispiel so aussehen:

Listing 1: main.c

```c
#include <stdio.h>
#include "uebung2.h"

int main(int argc, char* argv[])
{
  double result;
  int test_array[] = {1, 7, 5, 23, 42, -13, 28};
  int test_array_length = sizeof(test_array)/sizeof(int);

  result = min(test_array, test_array_length);
  printf("min: %0.0f\n", result);

  result = max(test_array, test_array_length);
  [...]

  return 0;
}
```

1. Implementieren Sie die Funktion `min()` in der Datei *uebung2.c*. In der Funktion soll das kleinste Element ermittelt und zurückgegeben (`return`) werden. Die prinzipielle Funktionsweise aller Funktionen dieser Übung ist im Programmablaufplan in Abbildung 1 dargestellt. Ist das übergebene Array leer, soll der Rückgabewert „Not a number“ sein. Lesen Sie dafür die Dokumentation zur Header-Datei *math.h* über `float`-Konstanten.

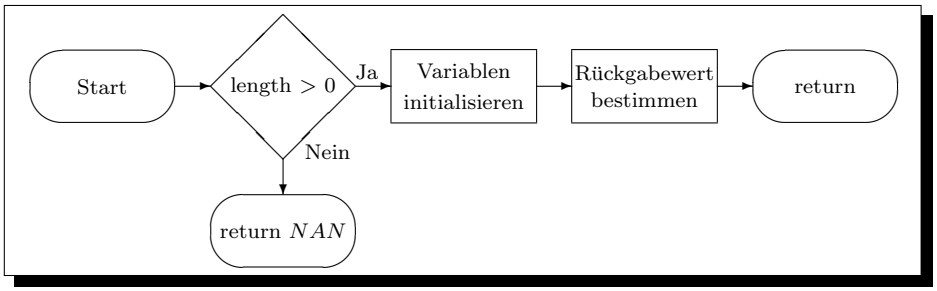

Abbildung 1: allgemeiner Programmablaufplan

Anmerkung: Da die Funktion `min()` und auch einige der folgenden Funktionen nur Integer-Werte zurückgeben können, würde man eigentlich als Rückgabewert `int` verwenden und nicht `double`. Da hier neben den einfachen Funktionen zusätzlich noch eine Fehlerprüfung durchgeführt werden soll, wurde der Typ `double` gewählt, um mit `NAN` Fehler zu signalisieren. Dies wäre auch möglich indem man einen weiteren Parameter an die Funktion übergibt, in der der Fehlercode zurückgegeben wird. Hierfür sind allerdings Zeiger (Pointer) notwendig, deshalb wurde hier darauf verzichtet, da Sie diese in der Vorlesung noch nicht behandelt haben.

2. Implementieren Sie die Funktion `max()`. Die Funktion soll das größte Element ermitteln und zurückgeben (`return`). Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist.

3. Implementieren Sie die Funktion `min_index()`. Die Funktion soll den Index des kleinsten Elements ermitteln und zurückgeben (`return`). Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist.

4. Implementieren Sie die Funktion `max_index()`. Die Funktion soll den Index des größte Elements ermitteln und zurückgeben (`return`). Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist.

5. Implementieren Sie die Funktion `sum()`. Die Funktion soll die Werte aller Elemente aufsummieren und das Ergebnis zurückgeben (`return`). Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist.

6. Implementieren Sie die Funktion `average()`. Die Funktion soll den Mittelwert (arithmetisches Mittel) aller Array-Elemente berechnen und zurückgeben (`return`). Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist.

7. Implementieren Sie die Funktion `stddev()`. Diese Funktion soll die Standardabweichung (standard deviation) einer gegebenen Menge (Array) berechnen. Auch hier soll der Rückgabewert „Not a number“ lauten, wenn das Array leer ist. Die Standardabweichung S ergibt sich aus:

$$S = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (X_i - \bar{X})^2}$$

Beachten Sie, dass der Laufindex i in der Summenformel bei „1“ beginnt und bei n endet. Der erste Index bei einem C-Array ist aber „0“! Die Quadratwurzel können Sie mit der Funktion `sqrt()` berechnen (siehe Dokumentation *math.h*).

Wenn Sie Probleme damit haben die Berechnung der Standardabweichung zu implementieren, nutzen Sie den Programmablaufplan in Abbildung 2 als Hilfestellung.

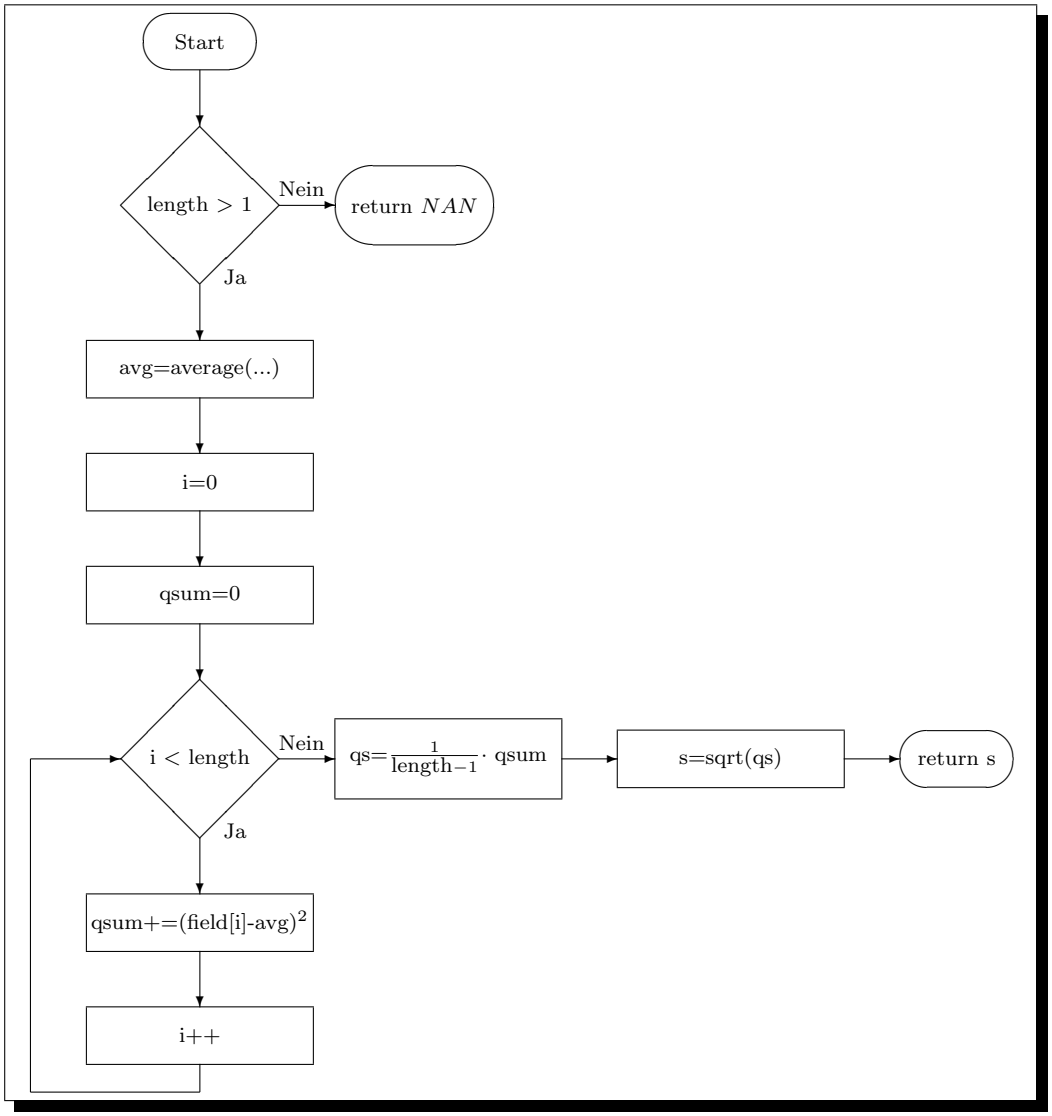

Abbildung 2: Programmablaufplan zur Berechnung der Standardabweichung