

## Übungsblatt 13: Einführung in C 2

25. Januar 2012

### Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Dienstag, 31.1., 13:00** für die Übungsgruppen am Donnerstag
- **Donnerstag, 2.2., 13:00** für die Übungsgruppen am Montag und Dienstag

Die Lösungen solltest Du in eine Kopie der Datei `/share/Courses/CG2011/13/vorlage13.txt` einfügen. Zur Abgabe schickst Du die Lösungsdatei im Anhang einer Email an Deinen Tutor.

### Aufgabe 13.1: Statistik von Datenreihen (5 Punkte)

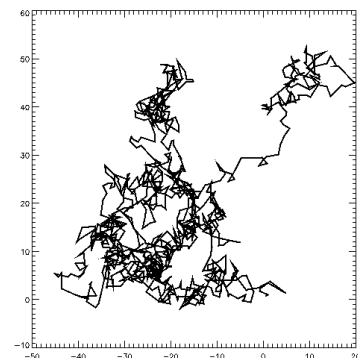
Im Verzeichnis `/share/Courses/CG2011/13` liegt die Datei `data.dat`, die Messdaten einer Simulation enthält, sowie den Anfang eines C-Programmes `stats.c`. Gesamtaufgabe ist es, das Programm so zu ergänzen, dass es die Datenreihen einliest, Mittelwerte, Minima und Maxima der einzelnen Datenreihen berechnet und dann zusammen mit ihrem Namen ausgibt (der Name der Datenreihe steht in der ersten Zeile von `data.dat`). Gehe dabei wie folgt vor:

- 13.1.1 (1 Punkte) Definiere die Datenstruktur `struct datacol` zum Speichern der Daten. Es soll den Namen (als `char[20]`) und die eigentlichen Daten der Reihe (als `float*`) enthalten.
- 13.1.2 (2 Punkte) Schreibe eine Funktion `read_data()`, die die Daten von der Standardeingabe mit Hilfe der Funktion `scanf()` in die Datenstrukturen einliest. Die Datenreihen selbst sollen dynamisch (d.h. mit `malloc()` bzw. `realloc()`) im Speicher angelegt werden, damit Datenreihen beliebiger Länge damit verarbeitet werden können.
- 13.1.3 (2 Punkte) Schreibe drei Funktionen `compute_min()`, `compute_max()` und `compute_mean()` die für jeweils eine Datenreihe den Mittelwert, Minima und Maxima berechnen.

Füge das fertige Programm in die Lösungsdatei ein.

### Aufgabe 13.2: Zufallspfade (5 Punkte)

Ein sogenannter *Zufallspfad* (*“random walk”*) ist ein wichtiges Konzept in der statistischen Physik, das beispielsweise die Diffusion von größeren Molekülen in Wasser (Brown'sche Molekularbewegung) oder auch die Konformation von Polymeren beschreibt. Wenn ein Teilchen in jedem Zeitschritt einen Schritt in eine zufällige Richtung macht, dann läuft das Teilchen auf einem Zufallspfad. Die nebenstehend Abbildung veranschaulicht einen solchen Pfad in zwei Dimensionen. Von größerem Interesse ist dabei, wie weit das Teilchen nach  $N$  Schritten gekommen ist. Diese Frage lässt sich sehr gut mit Hilfe eines Computers beantworten. In dieser Aufgabe sollst Du ein Programm schreiben, dass die Länge von solchen Zufallspfaden misst.





Dabei beschränken wir uns auf eindimensionale Pfade, das Teilchen macht also zufällig einen Schritt nach rechts oder nach links. Die Länge des Schrittes soll zufällig zwischen 0 und 1 liegen. Ein einzelner Zufallspfad umfasst  $N$  Schritte. Die Entfernung des Teilchens vom Ausgangsort nach  $N$  Schritten kann man mathematisch also wie folgt beschreiben:

$$r = \sum_{i=0}^N \xi_i$$

wobei  $\xi_i$  eine gleichverteilte Zufallszahl zwischen  $-1$  und  $1$  ist.

Wenn nun  $M$  Zufallspfade von jeweils  $N$  Schritten generiert werden, und jeweils die Entfernungen zum Ausgangsort  $r$  berechnet werden, dann kann man den sogenannten *RMSD* (*root mean square deviation*, die Wurzel der „mittlere quadratische Abweichung“) wie folgt berechnen:

$$d = \sqrt{\sum_{j=0}^M r_j^2}$$

wobei  $r_j$  die Entfernung des Teilchens vom Ausgangsort im  $j$ -ten Zufallspfad ist.

- 13.2.1 (4 Punkte) Schreibe ein Programm, das den RMSD  $d$  von jeweils  $M = 100000$  Zufallspfaden mit  $N \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$  Schritten berechnet und danach die Anzahl der Schritte und  $d$  in tabellarischer Form wie folgt ausgibt:

```
> ./rw
100 5.764953
200 8.178830
.
.
.
```

Füge das fertige Programm in die Lösungsdatei ein.

### Hinweise

- Schreibe für die Aufgabe am besten zwei C-Funktionen: eine Funktion erzeugt einen Zufallspfad von  $N$  Schritten und gibt  $r$  zurück (z.B. `double do_rw(int N);`), eine andere Funktion ruft diese Funktion dann  $M$  mal auf und berechnet damit den RMSD wie oben für  $M$  Zufallspfade von jeweils  $N$  Schritten (z.B. `double compute_rmsd(int M, int N);`). Das Hauptprogramm ruft diese Funktion dann in einer geeigneten Schleife auf und kümmert sich um die Ausgabe.
- Denke dabei an die Befehle zum Kompilieren und zum Generieren von Zufallszahlen vom letzten Übungsblatt!
- 13.2.2 (1 Punkte) Verwende `gnuplot` oder Python und `matplotlib`, um die Meßdaten zu plotten. Speichere den fertigen Plot in einem geeigneten Format und gib ihn zusammen mit der Lösungsdatei ab. Kannst Du ahnen, welche funktionale Form die Kurve hat? Entspricht das Deinen Erwartungen?