

Übungsblatt 1

C/C++ “Zum Aufwärmen”

(10.04.2014)

Wiederholung

- | | |
|---|--|
| 1. Was muss in einem C/C++-Programm auf jeden Fall im Quelltext stehen? | 2. Wie “loopt” man über das gesamte Array <code>int m[n]</code> korrekt? |
| <input type="checkbox"/> <code>#include</code> | <input type="checkbox"/> <code>for (int i=1; i<=n ; i++)</code> |
| <input type="checkbox"/> <code>main() { }</code> | <input type="checkbox"/> <code>for (int i=0; i<n ; i++)</code> |
| <input type="checkbox"/> <code>return</code> | <input type="checkbox"/> <code>for (int i=1; i<n ; i++)</code> |
| <input type="checkbox"/> <code>;</code> | <input type="checkbox"/> <code>for (int i=0; i<=n ; i++)</code> |

1. Aufgabe *Radiusberechnung* ★

Schreiben Sie ein C++-Programm zur Berechnung des Radius' eines Sterns aus seiner Leuchtkraft L_* , seiner Effektivtemperatur T_{eff} und der Stefan-Boltzmann-Konstanten σ_{SB} gemäß

$$L_* = 4 \pi \sigma_{\text{SB}} R_*^2 T_{\text{eff}}^4, \quad (1)$$

das den Nutzer nach der Eingabe von $\log L_*/L_\odot$ und T_{eff}/K fragt und daraus den Sternradius in R_\odot berechnet. Die Effektivtemperatur der Sonne beträgt $T_{\text{eff}} = 5778 \text{ K}$.

Zusatz: Versuchen Sie ungültige Eingaben abzufangen, also z.B. negative Temperaturen (einfach ★) oder Buchstaben (schwieriger ★★).

2. Aufgabe *Programmieren vs. Denken* ★★

Folgendes zahlentheoretisches Rätsel kann man selbstverständlich durch bloßes Nachdenken lösen. Alternativ werden wir hierfür eine kurzes C++-Programm zum Lösen schreiben.

In einem fernen Königreich verfügt der Herrscher anlässlich seines Thronjubiläums eine etwas exzentrische Form der Amnestie. Derzeit sind alle 100 Zellen des königlichen Kerkers mit je einem Gefangenen belegt. Zur Bewachung sind 100 Wärter angestellt. Der erste Wärter schließt jede Zelle auf. Dann kommt der zweite Wärter und schließt jede zweite Zelle wieder zu. Der dritte Wärter schließt jede dritte Zelle usw. Jeder i -te Wärter wird also jede i -te Zelle schließen. Dabei gilt: War die Zelle vorher offen, so ist sie nach dem Schließen zu und war sie vorher verschlossen, so ist sie anschließend offen. Ist die Zelle am Ende offen, so ist der Gefangene frei.

- Wieviele Gefangene sind am Ende frei?
- Wie oft dreht sich ein Schlüssel im Schloss der Zelle Nr. 100?

Hinweise: Es bietet sich die Verwendung eines Arrays `bool tuer[n]` an. Der Wert einer Booleschen Variablen `b` kann mittels `!b` invertiert werden.

3. Aufgabe *PostScript selbstgemacht* ★★ ★★

Wir wollen mithilfe eines C++-Programms eine PostScript-Datei (**.ps**) erzeugen, die einen n -zackigen Stern (bzw. Polygon) mit Umkreis enthält. PS-Dateien sind (ASCII)-Textdateien, die - ähnlich einem bash-Skript - Befehle enthalten, die vom PS-Viewer, z.B. **gv**, interpretiert werden. Unser C++-Programm sollte mittels Funktionen strukturiert werden.

- a) Funktion `openPS(char *name)` zum Anlegen einer PS-Datei **name** und Schreiben des Dateianfangs: ¹

```
%!PS-Adobe-1.0
gsave
```

- b) Funktion `closePS()` zum Schließen der PS-Datei und vorherigen Schreiben des Dateiendes:

```
showpage grestore
```

- c) Funktion `linePS(punkt from, punkt to)` um in PostScript eine Linie von (x_1, y_1) nach (x_2, y_2) zu zeichnen mittels:

```
x_1 y_1 moveto
x_2 y_2 lineto stroke
```

wobei **x_1** usw. *Pixel*-Koordinaten auf der PS-Seite sind. (0,0) liegt in der unteren linken Ecke der Seite.

- d) Funktion `circlePS(punkt p, float radius)` zum Zeichnen eines Kreises um (x, y) mit Radius r in PostScript:

```
x y r 0 360 arc stroke
```

- e) Die Position der i -ten Ecke des Sterns/Polygons lässt sich z.B. so berechnen:

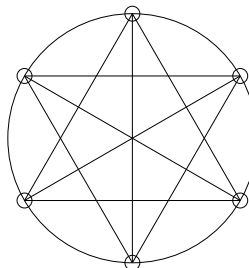
```
phi = ( (float) (i+1) - r0 ) * 2. * M_PI / (float) n ;
x = x0 + r0 * cos(phi) ;
y = y0 + r0 * sin(phi) ;
```

Der Mittelpunkt des Sterns könnte z.B. bei $x_0 = 0.7$, $y_0 = 0.8$ liegen, der Radius des Umkreises $r_0 = 0.5$ betragen.

Dabei müssten x und y durch Multiplikation mit einem Faktor (z.B. `float scale = 400 ;`) noch in Pixel umgerechnet werden.

- f) Struktur `struct punkt { float x , y ; }` ; für die Eckpunkte, d.h. es sollte im Programm dann ein entsprechendes Array `punkt[n]` angelegt werden.

- g) *Zusatz:* Die Ecken des Polygons/Sterns können noch jeweils mit einem kleinen Kreis verziert werden:



¹Der Einfachheit halber dürfen Sie ausnahmsweise ein *globales ofstream*-Objekt verwenden.