

Aufgabe 1: FORTRAN Übungsaufgabe

Dieses Übungsblatt umfasst Aufgaben zur Einarbeitung in die FORTRAN Programmierung und die Anwendung von Makefiles. Sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste:

`PPG-14@wr.informatik.uni-hamburg.de`

Aufgabe 1A: Abzählen im Kreis (60 Punkte)

Die zu programmierende Aufgabe umfaßt folgende Ausgangssituation:

Von 32 Personen wird ein Kreis gebildet. Eine Person wird als Nummer 1 identifiziert und dann wird von dieser Person ab abgezählt. Dabei verläßt jede dritte Person den Kreis und der Kreis wird ohne die fehlende Personen wieder geschlossen und ab der nächsten Person wieder ab 1 weitergezählt bis 3.

Die Aufgabe besteht darin, die Reihenfolge der Personen zu bestimmen, die den Kreis verlassen, in Bezug auf die ursprünglichen Aufstellung im Kreis. Die Reihenfolge der ausgeschiedenen Personen soll dargestellt werden bis auch die letzte Person aus der ursprünglichen Aufstellung im Kreis ausgezählt wurde.

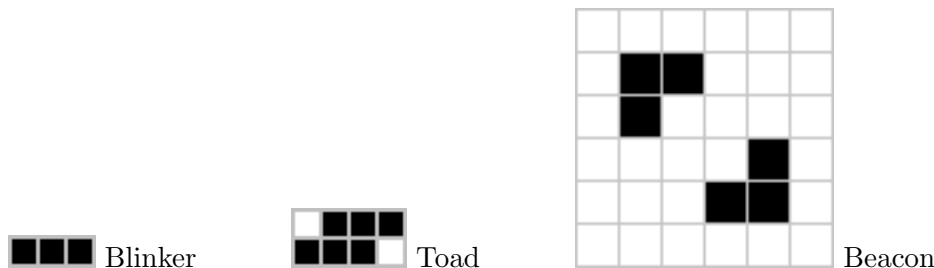
Die geforderte Ausgabe ist eine Liste der Nummern der ausgeschiedenen Personen, z.B. 3 6 9 (letzte ausgeschiedene Person)

Aufgabe 1B: Game of Life (180 Punkte)

In der zweite Aufgabe soll das Konzept von Conway's "Game of Life" in einem Programm umgesetzt werden. Für nähere Informationen zum Game of Life, siehe

http://en.wikipedia.org/wiki/Conway's_Game_of_Life

Hierbei sollen die "Lebenszyklen" der verschiedene Muster mit 10 Iterationen durchlaufen werden.



Dazu soll eine Matrix vom Typ Logic erstellt werden, die als Initialisierung mit 'false' belegt wird. Danach werden für die Muster der Figures Blinker, Toad und Beacon (s.u.) entsprechend der schwarzen Kästchen Werte in der Matrix mit dem Wert 'true' belegt. Die Berechnung soll auf einem Feld der Größe 30 X 20 erfolgen.

Zur Entwicklung des Lebenszyklus in der jeweils nächsten Iteration werden folgende Abfragen gestellt:

1. Hat eine lebende Zelle (schwarzes Kästen, in der Matrix mit 'true' belegt) weniger als 2 Nachbarn, so stirbt sie an Unterbevölkerung. D.h. sie wird in der nächsten Iteration auf 'false' gesetzt.
2. Hat eine lebende Zelle genau 2 oder 3 Nachbarn, so überlebt sie in der Iteration. D.h. der Wert 'true' bleibt erhalten.
3. Hat eine lebende Zelle mehr als 3 Nachbarn, so stirbt sie an Überbevölkerung.
4. Hat eine tote Zelle (weißes Kästen, in der Matrix mit 'false' belegt) genau 3 Nachbarn, so wird diese zu einer lebenden Zelle. D.h. sie wird in der nächsten Iteration auf 'true' gesetzt. Ansonsten bleibt die Zelle tot.

Es sollen die "Lebenszyklen" der verschiedene Muster mit 10 Iterationen durchlaufen und die Muster entsprechend dargestellt werden. Für die Ausgabe stellen wir das Programm Glider Movie auf unserer Webseite zur Verfügung, dessen Ausgangsroutine Unicode-Zeichen verwendet um eine gute lesbare Darstellung zu erreichen. Ihr dürft beliebige Teile von Glider Movie wiederverwenden, die **Subroutine printTwoDLogical** müsst Ihr verwenden.

Aufgabe 1C: Erstellen eines Makefiles (60 Punkte)

Die Datei **all-in-one.f95** soll in vier separate Dateien zerlegt werden. Es soll ein Makefile erstellt werden, mit dem das Programm kompiliert ('make') und auch ausgeführt werden kann ('make run'). Alle (!) Abhängigkeiten sollen korrekt berücksichtigt werden. Das heißt, wenn eine Datei neu abgespeichert wird, sollen bei einer erneuten Kompilation mit make nur die Datei selbst und die davon abhängigen Dateien neu übersetzt werden. Die nicht neu übersetzten Programmteile werden beim Linken wieder eingebunden und sollen ein lauffähiges Programm ergeben.

Abgabe

Die auf dem Cluster lauffähigen FORTRAN Programme sollen bis zum Dienstag den 15.4.2014 geschickt werden an:

ppg-abgabe@wr.informatik.uni-hamburg.de

Dabei ist zu beachten:

1. **NUR den Quellcode** schicken,
2. für **jede Aufgabe ein separates Verzeichnis anlegen** und
3. alles **als komprimiertes Archiv .tgz oder zip** schicken! D.h. es soll wirklich nur **ein einzelnes Archiv** geschickt werden!

Als Subject im Kopf der Mail bitte die Angabe: PPG-14 Blatt1 und die Liste der Familiennamen der Personen in der Übungsgruppe.