

## Aufgabe 3: FORTRAN und Makefile Übungsaufgaben

Dieses Übungsblatt umfasst eine Aufgaben zur weiteren Einarbeitung in die FORTRAN Programmierung und eine Aufgabe mit praktischen Übungen zum Thema Makefiles.

Wie gehabt, sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste:

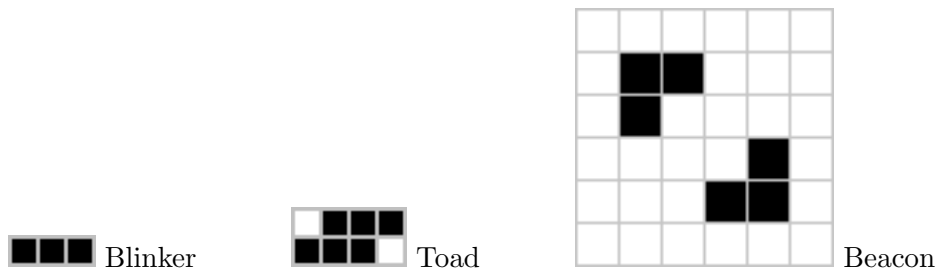
`PPG-13@wr.informatik.uni-hamburg.de`

### Aufgabe 3A: Game of Life (180 Punkte)

In dieser 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](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)

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



Quelle: [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)

Dazu soll eine Matrix vom Datentyp Logical erstellt werden, die als Initialisierung mit den Werten "false" vorbelegt wird. Danach werden für die Muster der Figuren Blinker, Toad und Beacon (s.u.) entsprechend der schwarzen Kästchen Werte in der Matrix mit dem Wert "true" belegt. Der Programmablauf soll auf einem Feld der Größe 30 X 20 erfolgen.

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

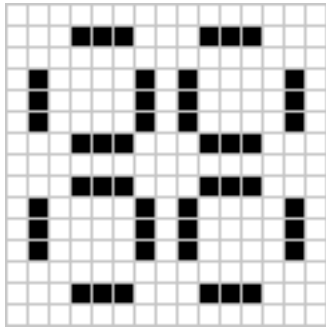
1. Hat eine lebende Zelle (schwarzes Kästchen, in der Matrix mit "true" belegt) weniger als 2 Nachbarn, so stirbt sie an Unterbevölkerung. D.h. sie wird in der nächsten Iteration auf "false" gesetzt.
2. Hat eine lebende Zelle genau 2 oder 3 Nachbarn, so überlebt sie in der Iteration. D.h. der Wert "true"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

Die geforderte Ausgabe ist die Darstellung der berechneten Matrix zu jeder Iteration (insgesamt 10) auf dem Bildschirm.

Dazu sollte das Write Statement `WRITE (*,'(30L1)') Game_Array(1:30,1:20)` (in diesem Beispiel wird die Logic Matrix `Game_Array` angelegt) benutzt werden. Als Überschrift für jede Matrix ist die Nummer der Iteration anzugeben, am Ende der Matrix soll eine Leerzeile stehen.

### Bonus-Aufgabe 3B: Game of Life - Pulsar (30 Punkte)

Als Zusatzaufgabe kann für das Game of Life zusätzlich der Pulsar dargestellt werden.



Pulsar

Quelle: [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)

Dazu sind die vorherigen Muster auszublenden (If Abfrage) und das Muster für den Pulsar (s.o.) auf der Matrix abzubilden und ebenfalls über 10 Iterationen darzustellen.

### Aufgabe 3C: 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 23.4.2013 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-13 Blatt3 und die Liste der Familiennamen der Personen in der Übungsgruppe.