**NADA, KTH** **Betygsgradering:** D-A
2D1312 2006-2007 (python)

# 122 Livet hos matematiska organismer

Läs anvisningarna på kurshemsidan!!!

**Varudeklaration:** Datastrukturer, filhantering.

En familj matematiska organismer består av celler (rutor) i ett rutnät som kan beskrivas av en matris. Cellerna föds och dör enligt bestämda levnadsregler och vi vill undersöka hur familjens fortbestånd förändras under ett antal generationer. Varje cell har åtta grannar:

```
1 2 3
4 * 5
6 7 8
```

**Överlevnad**

- Varje cell med två eller tre levande grannar kommer att överleva till nästa generation.
- En cell med mer än tre grannar dör av överbefolkning.
- En cell med mindre än två grannar dör av ensamhet.

**Födelse**

- En tom cell som har exakt tre grannar kommer att födas och bli en levande cell i nästa generation.

Skriv ett program som först läser in antalet önskade generationer samt storleken på matrisen. För att slippa tråkiga och tidsödande inmatningar ska du lägga upp filer med cellernas koordinater. En fil kan se ut så här:

```
Format: x-koord y-koord
============================================================
2 3
2 4
2 5
 ⋮
```

Programmet ska kontrollera att koordinaterna ligger inom tillåtet intervall. Användaren skall även kunna bestämma hur ofta han/hon vill se processens fortskridande. Användaren matar in ett heltal, ex. 4 , vilket tolkas som att användaren vill se var 4:e generation. Programmet skall vid en sådan presentation rita ut matrisen och dess levande innevånare på ett illustrativt sätt. Det är lämpligt att lägga in en pausfunktion så att användaren hinner se alla utritningar, t.ex. att användaren trycker retur innan nästa bild ritas upp.

Presentationen kan enklast ske via ascii-grafik, dvs vanliga tecken och blanka skrivs ut så att de tillsammans ser ut som en enkel figur.

**Tips:** Antag att du har valt att använda dig av en 15x15-matris. För att undvika vissa konstigheter ute i kanterna kan du då jobba med en 17x17-matris, detta för att kunna införa fiktiva celler i 0:te och 16:e position (så att alla verkliga celler får 8 grannar).

Matriserna kan lämpligen bestå av booleska element, som visar TRUE om en ruta innehåller en levande cell och FALSE med en död dito. Skriv en funktion, nygeneration, med två matriser som parametrar, levdenyss och levernu. I denna undersöks varje av levdenyss rutor enligt de ovanstående reglerna och en ny matris, levernu kan skapas (den nästkommande generationen).

Exempel på starttillstånd som ger en trevlig fortsättning:

```
18 levande celler:
----------
----------
---*--*---
---****---
--*----*--
--*-**-*--
--*----*--
---****---
----------
----------
```

```
CHESHIREKATT. Utskrift varje generation.
Lägg katten mitt i rutnätet. Efter några
generationer är bara leendet kvar.
```

```
5 levande celler:
-*--------
--*-------
***-------
----------
----------
 GLIDARE.
Vid utskrift av var 4:e generation
glider denna figur snett nedåt.
```

```
9 levande celler:
----*-----
-----*----
*----*----
-*****----
----------
 RYMDSKEPP.
Flyttar sig horisontellt om
utskrift görs var 4:e generation.
```

**Extrauppgift, betyg A:** Gör ett grafiskt gränssnitt, där förloppet ritas ut i ett rutnät istället. Lägg också till två knappar och ett textfält; En av knapparna ska stega bilden till nästan utritning, den andra ska rensa matrisen och i textfältet ska användaren mata in hur många generationer som ska gå mellan utritningarna.

**Extrauppgift, betyg A:** Gör så att man kan leka gud med musen, dvs att man kan skapa och/eller döda celler genom att trycka på rutorna.

**Extrauppgift, betyg A:** Lägg till en start- och en stoppknapp, och låt programmet stega fram med tex. 2 bilder per sekund.

**Datafiler och hjälpfiler:** `cheshirekatt.txt`
`glidare.txt`
`rymdskepp.txt`