**NADA, KTH** **Betygsgradering:** Grunduppgift lätt, betyg E-A
DD1310/DD1314 våren 2013 (Python)

# 104 Bulgarisk patiens

Läs anvisningar och betygsregler på kurshemsidan!!!

**Varudeklaration:** Datastrukturer. Sortering.

Regler för bulgarisk patiens:

- En del av en vanlig kortlek (minst 2 kort, högst 52) fördelas godtyckligt i några högar på bordet (valören på korten är ointressant i detta spel).
- Ett kort tas från varje hög och dessa får bilda en ny hög. Upprepa detta tills det går att avgöra att patiensen har gått ut (se nedan).

Exempel: Vi väljer följande tre högar (totalt 10 kort):
4 4 2
Efter ett steg har vi fyra högar:
3 3 1 3
Efter ytterligare ett steg försvinner en hög:
2 2 2 4 ⇒ 2 2 2 4
För enkelhetens skull skriver vi fr.o.m. nu högarna i storleksordning. I nästa steg får vi:
4 3 1 1 1
I nästa steg försvinner tre högar och en ny kommer till:
5 3 2
Inga högar försvinner, en kommer till. Vi får:
4 3 2 1
Nästa steg blir faktiskt också:
4 3 2 1
Likaså nästa...
4 3 2 1
osv...

Vi har alltså nått ett stabilt läge och patiensen har gått ut! Skriv ett program som läser in startfördelningen och som sedan skriver ut stegen fram till att patiensen har gått ut eller maximalt 25 steg. Högarna skall skrivas ut i storleksordning (inmatningen ska inte behöva vara det!). Startfördelningens antal kort måste kontrolleras ($2 \leq antal \leq 52$). Vid felaktig inmatning ska felmeddelande skrivas ut och användaren ska mata in en ny fördelning. Programmet ska även ge en kommentar om patiensen har gått ut eller ej.

**Extrauppgift, betyg C:** Om en kortfördelning återkommer efter några steg har en cykel uppstått och man vet då att patiensen inte kommer att gå ut.

```
3 2
2 2 1
3 1 1
3 2
osv...
```

Modifiera ditt program så att det avbryter direkt om en cykel uppstår.

Programmet ska också innehålla en kontroll av användarens inmatning så att det verkligen är ett heltal som kommer in. Om så inte är fallet ska programmet be om ett nytt värde.

**Extrauppgift, betyg B:** Om vi låter användaren ange enbart det totala antalet kort (istället för fördelningen) så finns det många olika sätt att fördela korten i högar. Ordningen ska inte spela roll så vi kan skriva högarna i storleksordning. T ex skulle vi med fyra kort kunna få följande fördelningar:

```
4
3 1
2 2
2 1 1
1 1 1 1
```

Låt ditt program beräkna alla fördelningar för ett givet antal kort, visa dem och erbjud användaren att välja en fördelning som patiensen sedan läggs med.

Om antalet fördelningar är alltför stort för att visa på skärmen ska programmet tala om hur många fördelningar som finns (beräkna med en formel) och sedan slumpa fram en fördelning (för givet antal kort) som patiensen läggs med.

**Extrauppgift, betyg A:** Gör ett grafiskt användargränssnitt (GUI) med knappar och menyer som är kopplade till relevanta funktioner.