

# 104 Bulgarisk patients

P-uppgiften ska göras individuellt. Läs CSC:s hederskodex innan du börjar!

**Varudeklaration:** Datastrukturer. Sortering.

Regler för bulgarisk patients:

- En del av en vanlig kortlek (minst 2 kort, högst 52) fördelas godtyckligt i några högar på bordet (värdet 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 patienten 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  $\Rightarrow$  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 patienten har gått ut! Skriv ett program som läser in startfördelningen och som sedan skriver ut stegen fram till att patienten har gått ut eller maximalt 25 steg. Högarna skall skrivas ut i storleksordning (inmatningen ska inte behöva vara det!). Startfördelningens antal kort måste kontrolleras ( $2 \leq \text{antal} \leq 52$ ). Vid felaktig inmatning ska felmeddelande skrivas ut och användaren ska mata in en ny fördelning. Programmet ska även ge en kommentar om patienten har gått ut eller ej.

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

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

Modifera 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 A:** Föreslå en egen utökning av ditt program efter kontakt med [stene@kth.se](mailto:stene@kth.se)

# 132 Romerska årtal

P-uppgiften ska göras individuellt. Läs CSC:s hederskodex innan du börjar!

**Varudeklaration:** Stränghantering.

Gör ett program som översätter romerska årtal till arabiska (vanliga) årtal och tvärtom. Årtalen ska kunna variera från 1 till 3999.

Gör också en romersk miniräknare som kan addera romerska tal.

Programmet ska kunna tolka inmatningen utan menyval. Körexempel:

```
Välkommen till den romerska årtalskonverteraren Anno Domine!
```

```
Årtal: 1066
```

```
... dvs MLXVI
```

```
Årtal: 4711
```

```
Jag klarar tyvärr bara årtal från 1 till 3999.
```

```
Årtal: MDCCCLXXXVIII
```

```
... dvs 1888
```

```
Årtal: 1990
```

```
... dvs MCMXC
```

```
Årtal: XM
```

```
Det är inte ett korrekt romerskt tal.
```

```
Årtal: CMXC
```

```
... dvs 990
```

```
Årtal: XII + IX
```

```
... dvs XXI
```

```
Årtal:
```

```
Pax vobiscum önskar Anno Domine!
```

Följande tecken förekommer i romerska talen:

Romerska:	I	V	X	L	C	D	M
-----------	---	---	---	---	---	---	---

Arabiska:	1	5	10	50	100	500	1000
-----------	---	---	----	----	-----	-----	------

De romerska talen ordnas så att de största talen kommer först, de enda undantagen är dessa kombinationer:

Romerska:	IV	IX	XL	XC	CD	CM
-----------	----	----	----	----	----	----

Arabiska:	4	9	40	90	400	900
-----------	---	---	----	----	-----	-----

OBS! Inga andra specialfall behöver testas (trots att det i verkligheten finns sådana).

Bokstavsvärdena i kombinationen adderas till en summa, studera körexemplet ovan samt denna lista:

Romerska:	I	II	III	IV	V	VI	VII	VIII	IX	X	XI
-----------	---	----	-----	----	---	----	-----	------	----	---	----

Arabiska:	1	2	3	4	5	6	7	8	9	10	11
-----------	---	---	---	---	---	---	---	---	---	----	----

**Tips:** Programmet skall inte undersöka en massa specialfall. Regeln att de största talen kommer först gäller, med de enda undantagen givna ovan. Betrakta IV, IX etc som tal. Här är det risk för kodupprepning, så satsa på generella lösningar.

**Extrauppgift, betyg C:** Inför felkontroll av inmatade romerska årtal med följande enkla algoritm:

- Omvandla det romerska årtalet till ett arabiskt.
- Omvandla tillbaka detta arabiska årtal till ett romerskt årtal.
- Är detta romerska årtal identiskt med det som matades in från början?
- Om de är lika är årtalet korrekt, annars felaktigt.

**Extrauppgift, betyg A:** Föreslå en egen utökning av ditt program efter kontakt med [stene@kth.se](mailto:stene@kth.se)