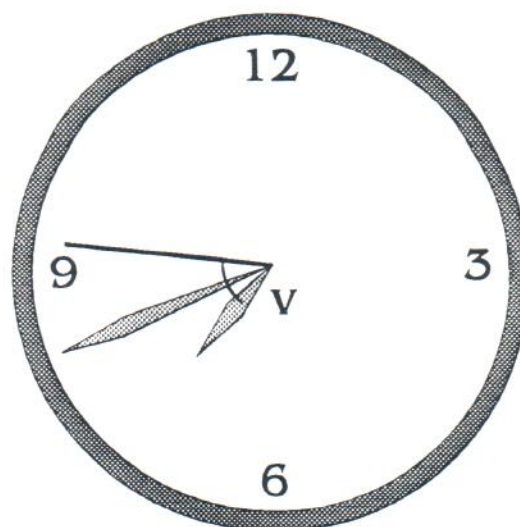


# Uppgift 1

Sekundvisaren på en klocka hoppar fram en sekund i taget. Minut- och timvisare däremot rör sig kontinuerligt över urtavlan. Vi tolvslaget står alla visarna över varandra. Vid vilket klockslag efter kl 12:00:00 och före kl 00:00:00 är visarna 'mest samlade'. Det vill säga när är vinkeln  $v$  i *fig 1*, så liten som möjligt? Skriv ett program som löser problemet.

**Indata:** -

**Utdata:** Vinkeln  $v$  (i grader) och den tidpunkt, på formen XX:XX:XX, då vinkeln  $v$  antar sitt minimum. Om det finns flera tidpunkter med samma minimum ska alla presenteras.



*fig 1. Timvisare och sekundvisare är vid denna tidpunkt längst ifrån varandra. Dessa två visares placering avgör därför storleken på  $v$ .*

# Uppgift 2

I bowling ska man, med klotet, försöka få ner alla 10 käglorna. Man har två slag på sig. Om man lyckas i första slaget kallas det *strajk* (det stavas faktiskt så i de svenska reglerna) och om alla källor fallit efter två slag kallas det *spärr*. Poängberäkningen i bowling är lite speciell och lämpar sig bra för ett program! Följande regler gäller för poängsättningen:

- ☐ En serie innehåller 10 rutor
- ☐ I varje ruta får man göra högst två slag
- ☐ Slår man ner alla tio käglorna (strajk) i första slaget får man 10 poäng plus lika många poäng, som antalet källor man kommer att slå ner i nästkommande två slag.
- ☐ Har man lyckats få ner alla käglorna efter två slag (spärr) får man 10 poäng plus lika många poäng, som antalet källor man kommer att slå ner i nästa slag.
- ☐ Finns källor kvar efter två slag, får man lika många poäng som det sammanlagda antalet nedslagna källor i de två slagen.
- ☐ Om man får strajk i *sista rutan* får man två extraslag och lika många poäng, i sista rutan, som antalet nedslagna källor i de tre sista slagen.
- ☐ Om man får spärr i *sista rutan* får man ett extraslag och lika många poäng, i sista rutan, som antalet nedslagna källor i de tre sista slagen.

| Bana | 1   | 2   | 3  | 4  | 5   | 6   | 7   | 8   | 9   | 10    | X / | 0 - | P   | Res |
|------|-----|-----|----|----|-----|-----|-----|-----|-----|-------|-----|-----|-----|-----|
|      | 9 / | 8 1 | X  | X  | X   | 8 / | X   | X   | 7 / | X X 8 | 6 3 | 1   | 220 |     |
|      | 18  | 27  | 57 | 85 | 105 | 125 | 152 | 172 | 192 | 220   |     |     |     |     |

Fig 2a. Ett bowlingprotokoll, där X står för strajk och / för spärr. Under P finns slutresultatet (i de smala kolumnerna i till vänster en del statistik). I serien har gjorts 16 slag.

I fig 2a finns ett bowlingprotokoll, där X står för strajk och / för spärr. Skriv ett program som frågar efter antalet käglor som slås ned i varje slag och som har en dialog liknande den i fig 2b

**Indata:** En fullständig bowlingserie, slag för slag. Se fig 2b.

**Utdata:** Den inmatade bowlingseriens slutpoäng.

```

RUTA 1 SLAG 1 ? 8
RUTA 1 SLAG 2 ? 2
RUTA 2 SLAG 1 ? 10
RUTA 3 SLAG 1 ? 9
RUTA 3 SLAG 2 ? 1
RUTA 4 SLAG 1 ? 6
RUTA 4 SLAG 2 ? 2
RUTA 5 SLAG 1 ? 10
RUTA 6 SLAG 1 ? 10
RUTA 7 SLAG 1 ? 10
RUTA 8 SLAG 1 ? 8
RUTA 8 SLAG 2 ? 2
RUTA 9 SLAG 1 ? 7
RUTA 9 SLAG 2 ? 3
RUTA 10 SLAG 1 ? 10
EXTRASLAG 1 ? 7
EXTRASLAG 2 ? 3
POÄNGEN BLEV 199

```

## Uppgift 3

Siffrorna i ett fyrsiffrigt tal, där inte alla fyra siffrorna är lika, sorteras, dels i stigande och dels i fallande ordning. Dessa bildar nya fyrsiffriga tal. Det mindre talet subtraheras från det större och resultatet behandlas som ett nytt fyrsiffrigt tal, möjligtvis inlett med nollor, som sorteras som det första. En ny subtraktion utförs på samma sätt och processen fortsätter tills subtraktionen ger resultat 6174. Det konstiga är nämligen att detta tal kommer att dyka upp förr eller senare. Fig 3 visar ett exempel där vi startar med talet 3332.

Skriv nu ett program som bestämmer största antalet varv som behövs för att, från ett tal mellan 1000 och 9998, nå fram till talet 6174 och hur många utgångstal som kräver detta maximala antal iterationer.

**Indata:** -

**Utdata:** Största antalet iterationer (varv), som något tal mellan 1000 och 9998 behöver för att nå fram till 6174.

Antal tal mellan 1000 och 9998, som behöver detta maximala antal iterationer.

fig 3

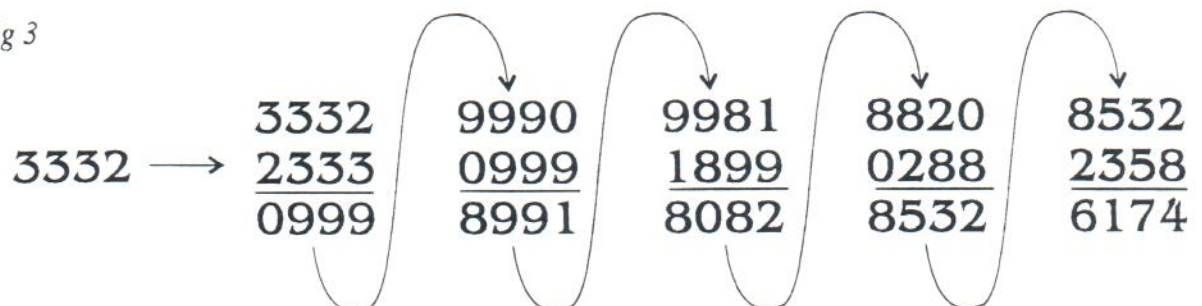


Fig 3. Talet 3332 behöver fem iterationer innan talet 6174 kommer fram. Självfallet ger startvärdet 6174:  $7641 - 1467 = 6174$  direkt



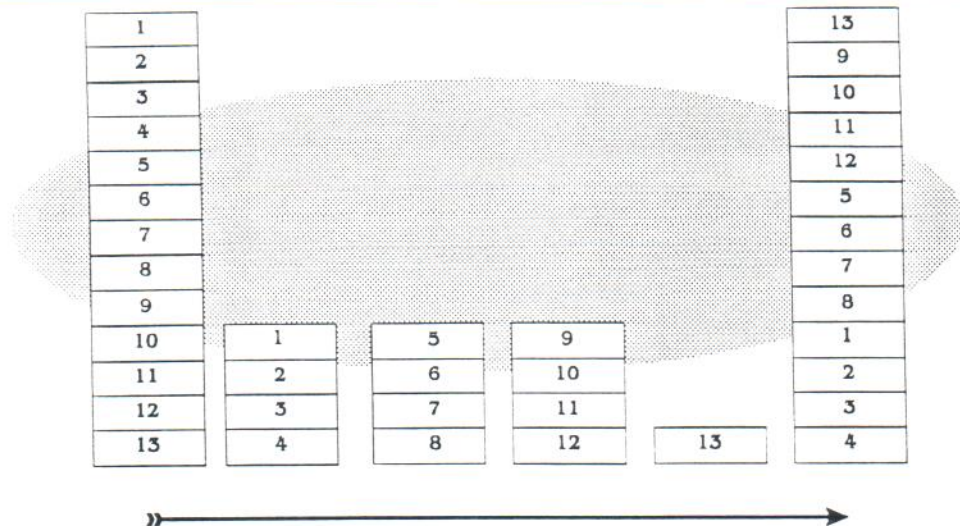


Fig 4. Med 13 marker i utgångshögen och 4 marker i varje liten hög. Först flyttas markerna 1 - 4 till en egen hög och därefter markerna 5 - 8 och 9 - 12. Marker nr 13 blir över och får bilda en egen hög. Ihopplockandet börjar sedan med den första lilla högen, 1 - 4 och därefter 5 - 8 högen 9 - 12 högen. Till sist läggs högen med enbart marker 13 på toppen. Nedbrytandet och uppbyggandet fortsätter tills markerna i den stora högen nått ursprungsplaceringen, 1 - 13 från toppen räknat.

## Uppgift 4

Medan kulan hoppar runt i den snurrande rouletten fingrar croupiern nervöst med spelmarkerna, enligt följande mönster, se fig 4. Från utgångshögen, traven till vänster, lyfter croupiern upprepade gånger av lika många marker och placerar dem i en rad högar intill. Den sista högen kan möjligen komma att innehålla färre marker än de andra. Därefter börjar han plocka ihop högarna igen, men nu i en annan ordning. När alla marker åter ligger i en trave påbörjar han uppdelandet i småhögar igen. Mönstret upprepas så gång på gång.

Skriv ett program som med vetskap om antalet marker i utgångshögen och antalet marker i småhögarna tar reda på hur många gånger ritualen måste utföras för att alla marker i den stora högen ska återfå samma placering som de ursprungligen hade.

**Indata:** Antalet marker,  $n$  i utgångshögen ( $n < 50$ )

Antal marker i småhögarna,  $h$  ( $h < n$ )

**Utdata:** Antal varv croupiern måste genomföra för att markerna i utgångshögen ska återfå sin ursprungsplacering.

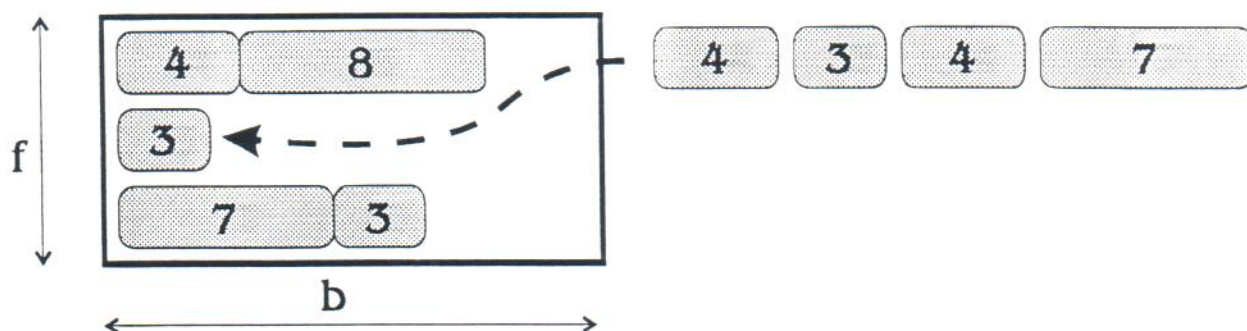
## Uppgift 5

En färja ska lastas från en bilkö. Färjan har en längd på  $b$  meter och är indelad i  $f$  filer (alla filer är  $b$  meter långa). Eftersom det finns bilar av olika längder i kön (endast hela meter) finns det olika sätt att lasta färjan. En bil som står i tur att köra upp på färjan kan dirigeras, av färjpersonalen, till någon av de  $f$  filerna, under förutsättning att det finns tillräckligt med plats i den filen. Se *fig 5*.

Skriv ett program som planerar hur bilarna ska fördelas mellan de olika filerna, så att färjan rymmer så många bilar som möjligt. Observera att en bil aldrig får bryta kön och köra om framförvarande bilar!

**Indata:** . Färjans längd,  $b$  meter (heltal  $b < 20$ ). Antalet filer,  $f$  ( $f < 5$ ). Antalet bilar  $n$ , i kön ( $n < 20$ ). Bilarnas längder i meter (heltal), givna i den ordning de återfinns i kön, från den första till den sista.

**Utdata:** Det maximala antalet bilar som får plats på färjan om den lastas på bästa sätt. Den sammanlagda sträckan av outnyttjad plats (i meter) i de  $f$  filerna.



*Fig 5. En typisk situation. Bilen som står på tur kan placeras i två olika filer. Vilket val ger den bästa lastningen av färjan?*

## Uppgift 6

För några år sedan gick ett TV-program med namnet *Tänk Till Tusen*. I ett av momenten i detta tävlingsprogram gällde det att utföra en aritmetisk beräkning med ett antal givna tal och komma så nära ett givet tresiffrigt tal som möjligt.

Förutom de tresiffriga 'måtalet' fick man sex heltal hämtade ur mängden: 1, 2, 3 .. 9, 10, 25, 50, 75, 100. Flera tal av samma valör kan förekomma.

Med hjälp av de fyra räknesätten, *addition*, *subtraktion*, *multiplikation* och *division*, gäller det sedan att komma så nära, helst ända fram till, *måtalet*. Det fanns dock några speciella regler, som man måste rätta sig efter.



- ☐ Beräkningarna görs en i taget och det handlar alltså aldrig om prioritetsregler och parenteser (beräkningarna utförs som om de vore gjorda av en 'dum' räknedosa). Se *fig 6*.
- ☐ Var och en av de sex talen får bara användas en gång, men man behöver inte använda alla sex.
- ☐ Delresultaten måste alltid vara heltal, det vill säga divisioner under resans gång måste 'gå jämnt ut'.
- ☐ Delresultaten får aldrig bli  $<0$ , något som skulle kunna inträffa vid oförsiktig subtraktion.

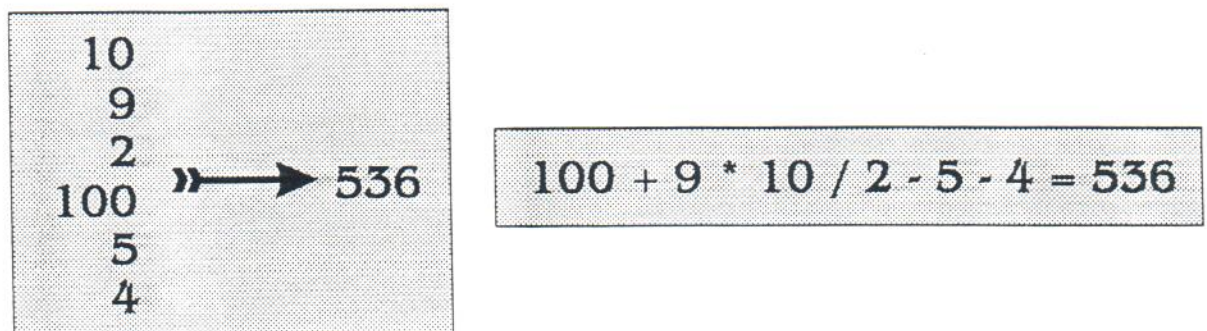
**Indata:** Först de sex talen som ska ingå i beräkningarna  
Sedan det tresiffriga måltalet.

**Utdata:** Antingen en lösning till problemet på formen

*tal operation tal operation tal ... tal = 'måltal'*

eller texten INGEN LÖSNING FUNNEN.

Observera att även om problemet har flera lösningar så räcker det att programmet finner en av dessa.



*Fig 6. Ett exempel på en sifferfråga från TV-programmet Tänk Till Tusen. Målet är att med hjälp av de fyra räknesätten och de sex talen till vänster om pilen räkna sig fram till talet 536.  $100 + 9 = 109$ ,  $109 * 10 = 1090$ ,  $1090 / 2 = 545$ ,  $545 - 5 = 540$ ,  $540 - 4 = 536$ .*