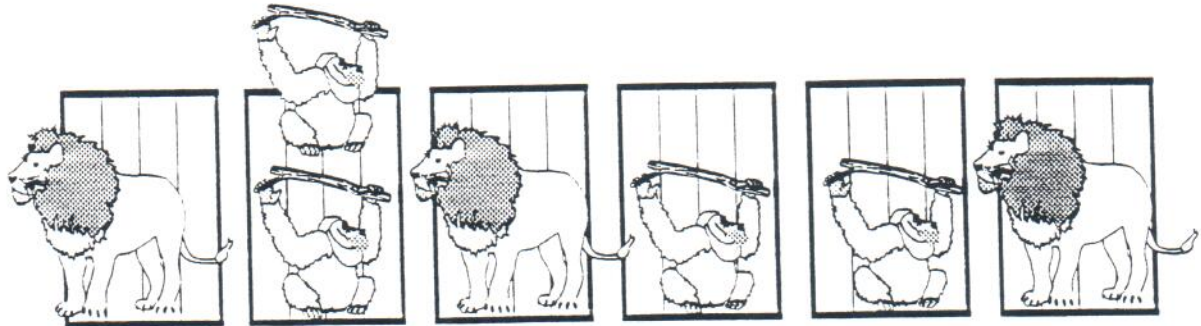


Uppgift 1

Givet ett antal *burar*, ett antal *lejon* och ett antal *apor*. Hur många arrangemang av djuren i olika burar finns det om:



figur 1

- ☐ Det får finnas högst ett lejon i en och samma bur
- ☐ Två lejon aldrig får sitta i två burar intill varandra
- ☐ Det får finnas högst två apor i en och samma bur
- ☐ En apa måste alltid ha sällskap, av en apa i samma bur eller minst en apa i intilliggande burar.

Skriv ett program som listar alla möjliga arrangemang och dessutom bestämmer antalet lösningar. Antal burar i testexemplen är <10.

Indata

En dialog där operatören får möjlighet att besvara följande frågor:

Antal burar: 6
Antal lejon: 3
Antal apor : 4

Utdata

En rad, på skärmen, för varje kombination. Exemplet ger följande resultat, följt av en rad som berättar hur många lösningar det finns:

```
Lejon  2_Apor  Lejon  2_Apor  Lejon  Tom
Lejon  2_Apor  Lejon  Tom      Lejon  2_Apor
Lejon  Tom     Lejon  2_Apor  Lejon  2_Apor
Lejon  2_Apor  2_Apor  Lejon  Tom     Lejon
Lejon  2_Apor  Tom     Lejon  2_Apor  Lejon
Lejon  1_Apa   1_Apa   Lejon  2_Apor  Lejon
Lejon  Tom     2_Apor  Lejon  2_Apor  Lejon
2_Apor  Lejon  2_Apor  Lejon  Tom     Lejon
2_Apor  Lejon  Tom     Lejon  2_Apor  Lejon
Tom     Lejon  2_Apor  Lejon  2_Apor  Lejon
Lejon  2_Apor  Lejon  2_Apor  Tom     Lejon
Lejon  2_Apor  Lejon  1_Apa  1_Apa  Lejon (se fig 1)
Lejon  2_Apor  Lejon  Tom     2_Apor  Lejon
Lejon  Tom     Lejon  2_Apor  2_Apor  Lejon
DET FINNS 14 LÖSNINGAR
```

Uppgift 2

På ett bord ligger ett antal travar med böcker. Varje dag tas en bok från varje trave. Dessa böcker bildar tillsammans en ny trave till höger om de andra, se *figur 1*. Om en trave blir tom skjuts travarna samman från höger.

Eftersom det finns ett ändligt antal böcker, kommer förr eller senare samma upplägg att återkomma. Din uppgift blir nu att skriva ett program som frågar efter första uppläggets utseende och sedan ta reda på hur lång tid det tar innan ett upplägg som tidigare funnits, återkommer.

I tabellen i *figur 2* visas hur utseendet av boktravarna varierar under 13 dagar. I de testexempel som valts är antalet böcker < 50 , antalet travar som mest < 15 och antalet dagar < 100 .

Indata:

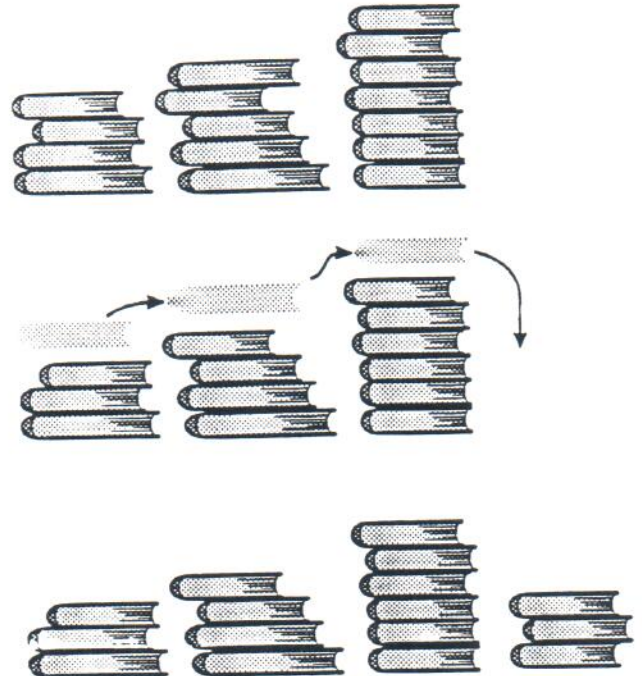
Exempel på dialog enligt exemplet ovan

```
ANTAL TRAVAR: 3
BÖCKER I TRAVE 1: 4
BÖCKER I TRAVE 2: 5
BÖCKER I TRAVE 3: 7
```

Utdata:

Resultatet skrivs ut som en mening

```
DAG 13 UPPKOM ETT UPPLÄGG, SOM
REDAN FÖREKOMMIT DAG 7
```



figur 1

Dag	Hög 1	Hög 2	Hög 3	Hög 4	Hög 5	Hög 6
1	4	5	7			
2	3	4	6	3		
3	2	3	5	2	4	
4	1	2	4	1	3	5
5	1	3	2	4	6	
6	2	1	3	5	5	
7	1	2	4	4	5	
8	1	3	3	4	5	
9	2	2	3	4	5	
10	1	1	2	3	4	5
11	1	2	3	4	6	
12	1	2	3	5	5	
13	1	2	4	4	5	

figur 2

Uppgift 3

Gallupundersökningarna har avslöst varandra den här hösten. Väljarsympatierna presenteras normalt i form av procentsatser.

En person, som frågade sin närmaste omgivning om hur de tänkte rösta i EU-valet fick till svar: 48% JA och 52% NEJ. Vilket är det minsta antal personer han kan ha tillfrågat, om procenttalen är avrundade till heltal?

Svaret är 21, fördelade på 10 JA (47.619%) och 11 NEJ (52.381%).

Undersökningen med 33 deltagare som gav 16 JA (48.485%) och 17 NEJ (51.515%) ger visserligen samma resultat men innehåller onödigt många tillfrågade.

Skriv ett program som bestämmer det minsta antalet deltagare i en väljarundersökning, där undersökningsresultatet är givet i procenttal avrundade till heltal.

Indata:

Programmet ska först ta emot uppgifter om *antalet grupper* och sedan efter den avrundade procentsatsen (heltalet) för var och en av grupperna. Summan av de inmatade procenttalen kan alltid antas bli 100. Antalet grupper är i testexemplen <7.

```
ANTAL GRUPPER: 4
PROCENT GRUPP 1: 28
PROCENT GRUPP 2: 39
PROCENT GRUPP 3: 13
PROCENT GRUPP 4: 20
```

Eftersom *avrundningsreglerna* kan anses oklara, kommer inga testdata, att väljas, där resultatet beror på om avrundningen av ett tal, mitt emellan två heltal ska avrundas uppåt eller nedåt (ex 10.5 till 10 eller 11)

Utdata:

Det tal som anger minsta antalet deltagare i undersökningen, där det finns en fördelning mellan grupperna, som omräknat till procent och avrundat till heltal, överensstämmer med indata.

UNDERSÖKNINGEN KAN SOM LÄGST INNEHÅLLA 46 PERSONER

Uppgift 4

Ett gammalt klassiskt problem lyder: Fyra män har lidit skeppsbrott på en öde ö där det inte finns något annat ätbart än kokosnötter.

De märkte snart att öns apor satte i sig väldiga mängder och för att själva slippa råka ut för kokosnötsbrist samlade de ihop alla nöterna i en stor hög.

Sedan de gått och lagt sig på kvällen vaknade emellertid snart en av de fyra igen och muttrade bistert för sig själv:

"Jag känner ju inte alls de tre andra, och har ingen aning om huruvida de är ärliga eller inte. Så det är nog säkrast att jag genast tar min andel av kokosnötterna"

Han gick ut och delade upp kokosnötterna i fyra lika stora högar, upptäckte att det blev tre över, som han kastade till aporna, gömde sin fjärdedel och samlade ihop de återstående i en hög, varefter han kröp till kojs igen.

Strax efteråt vaknade en annan av männen, funderade sig fram till samma slutledning som den förste och delade upp de återstående kokosnötterna i fyra lika stora högar, upptäckte att det blev tre över (dem kastade han åt aporna), gömde sin fjärdedel och samlade ihop de återstående i en hög igen, och kröp till kojs på nytt. De båda andra upprepade i tur och ordning samma förfarande. Nästa morgon var högen synbart mindre än kvällen innan, men eftersom de alla fyra led av en viss skamkänsla var det ingen som sade någonting, utan alla gick villigt med på det förslag som väcktes om att var och en skulle ta sin del och använda den efter eget gottfinnande.

Alltså delade de upp de återstående kokosnötterna i fyra lika delar (den här gången blev det inte en enda nöt över åt aporna) och tog hand om var sin fjärdedel.

Vilket är det minsta antal kokosnötter högen ursprungligen kan ha innehållit?

Skriv ett program som frågar efter antalet sjömän (≤ 7) och antalet kokosnötter som aporna fick vid de första delningarna (mellan 0 till antal *sjömän*-1) och som sedan beräknar det minsta antalet kokosnötter ($< 6\,000\,000$) som behöver finnas i högen från början för att proceduren ska bli möjlig. Tänk på att det måste finnas kokosnötter kvar i högen när dagen gryr!

Indata:

ANTAL SJÖMÄN: 4
TILL APORNA : 3

Utdata:

DET BEHÖVS 247 KOKOSNÖTTER I HÖGEN FRÅN BÖRJAN.

Uppgift 5

I figur 1 ser du en serietabell (med förklaringar), där fyra lag har mött var och en av de andra lagen en gång - spelat tre matcher. Ingen av matcherna har slutat med *samma resultat* och i ingen match har det gjorts *mer än fem mål*. Skriv ett program som utifrån en given tabell bestämmer slutresultatet i de sex matcherna! Observera att en hemmaseger med samma siffror som en bortaseger, till exempel hemmasegern 2-1 och bortasegern 1-2, betraktas som *samma resultat*.

Till samtliga testexempel finns exakt en lösning.

Lag	Antal matcher	Vunna	Oav gjorda	För lorade	Gjorda mål	Insläppta mål	Poäng
A	3	3	0	0	11	-	3
B	3	1	1	1	3	-	4
C	3	0	2	1	2	-	6
D	3	0	1	2	4	-	7

Figur 1. Laget A har vunnit turneringen genom att vinna alla tre matcherna, gjort 11 mål och släppt in 3. Poängen 6 erhålles genom 2 poäng för varje seger (gamla regler). Oavgjort ger 1 poäng

I. Jata

Programmet frågar efter fyra rader, där varje rad i sin tur innehåller sju olika data. Ett exempel på hur indata kan ges:

```
LAG A: 3 3 0 0 11 3 6
LAG B: 3 1 1 1 3 4 3
LAG C: 3 0 2 1 2 6 2
LAG D: 3 0 1 2 4 7 1
```

Observera att lagens namn alltid är A, B, C och D och dessutom står i denna ordning.

Utdata

Sex rader, där varje rad innehåller ett matchresultat, beteckning på två lag och motsvarande mål. Löser vi exemplet erhåller vi följande utdata

```
A - D 3-2
A - C 4-0
A - B 4-1
D - C 2-2
D - B 0-2
C - B 0-0
```

Den inbördes ordningen mellan matcherna har ingen betydelse.

Uppgift 6

Du har till ditt förfogande n ($n \leq 6$) stycken kugghjul, alla med ett utseende som i *figur 1*. För varje kugghjul finns data om antalet kuggar, både för den yttre och den inre kransen.

Två kugghjul a och b kan sammanfogas på tre olika sätt, se *figur 2*, och observera samtidigt att de båda inre kransarna aldrig direkt kan förenas.

Uppgiften består i att skriva ett program som utifrån uppgifter om kugghjulen bestämmer utseendet hos en kuggväxel som ger en önskad utväxling. I *figur 3* och *figur 4* ges två exempel, där den önskade utväxlingen är 7 respektive 196/125. För båda exemplen har kugghjulen i *figur 5* funnits tillgängliga.

Det är alltid hjulet längst till vänster som är det drivande. Utväxlingen definieras som det antal varv det sista hjulet snurrar då det drivande gör precis ett varv.

Alla kugghjulen behöver inte användas. Om flera lösningar ger samma resultat, ska den lösning med minsta antalet kugghjul presenteras. Av flera lösningar med samma antal kugghjul räcker det att ge en.

Indata:

Allt enligt exemplet nedan:

```

ANTAL KUGGHJUL: 5
KUGGHJUL 1 YTTRE: 10
KUGGHJUL 1 INRE : 6
KUGGHJUL 2 YTTRE: 25
KUGGHJUL 2 INRE : 12
KUGGHJUL 3 YTTRE: 14
KUGGHJUL 3 INRE : 8
KUGGHJUL 4 YTTRE: 30
KUGGHJUL 5 INRE : 21
KUGGHJUL 6 YTTRE: 28
KUGGHJUL 7 INRE : 15
ÖNSKAD UTVÄXLING
TÄLJARE: 196
NÄMNARE: 125
    
```

Den önskade utväxlingen ges som en kvot, med täljare och nämnare.

Utdata:

Kuggväxeln beskrivs hjul för hjul från vänster till höger. Resultatet från exemplet i *figur 4* blir

4 I Y 1 Y I 5 Y Y 2

Hjul nr 4's inre krans är kopplad till den yttre på hjul nr 1, och så vidare.

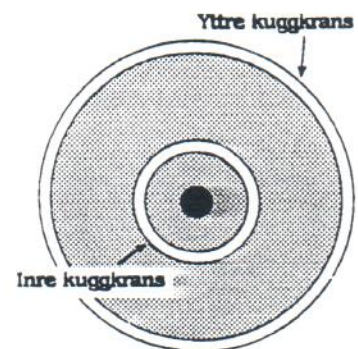


fig 1

Hjul nr	Yttre	Inre
1	10	6
2	25	12
3	14	8
4	30	21
5	28	15

fig 5

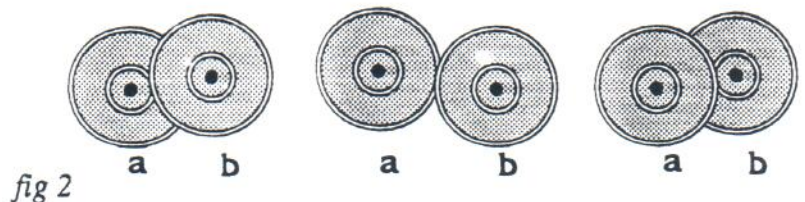


fig 2

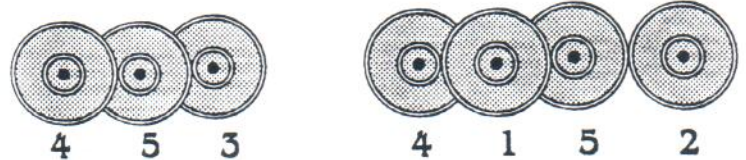


fig 3

fig 4