

UPPGIFT 1 – WILL ROGERS FENOMEN

Will Rogers (1879-1935) var en amerikansk komiker känd för bland annat följande citat:

"When the Okies left Oklahoma and moved to California, they raised the average intelligence level in both states."

Denna skenbara paradox, att flyttning av ett element från en mängd till en annan gör att medelvärdet ökar i båda mängderna, har därför fått namnet Will Rogers fenomen. Du ska skriva ett program som läser in två grupper A och B vardera bestående av minst två och högst tio positiva heltal och avgör huruvida det är möjligt att genom att flytta ett tal från den ena gruppen till den andra få medelvärdet att öka i båda grupperna och i så fall vilket tal som ska flyttas till vilken grupp. Om det finns flera möjligheter så räcker det att ange en av dem.

Körningsexempel:

```
Antal tal i grupp A? 3
Tal ? 3
Tal ? 1
Tal ? 2
Antal tal i grupp B ? 4
Tal ? 4
Tal ? 3
Tal ? 5
Tal ? 4
```

Flytta talet 3 till A.

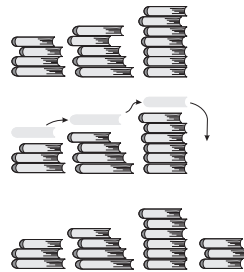
Ett andra körningsexempel:

```
Antal tal i grupp A ? 2
Tal ? 7
Tal ? 5
Antal tal i grupp B ? 2
Tal ? 4
Tal ? 6
```

Omöjligt!

Kommentar: I det första exemplet är medelvärdena 2 respektive 4 innan flyttning. Efter att talet 3 flyttats över från B till A är medelvärdena 2.25 respektive 4.333... I det andra exemplet kan fenomenet inte uppkomma. Om man t.ex. flyttar talet 5 från A till B så ökar visserligen medelvärdet i grupp A men medelvärdet i grupp B förblir oförändrat.

UPPGIFT 2 – TRAVAR MED BÖCKER



FIGUR 1. Första dagen plockas en bok från var och en av de tre travarna. Dessa tre böcker bildar en ny trave till höger om de andra.

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 tar reda på hur lång tid det tar innan ett upplägg som tidigare funnits, återkommer.

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	

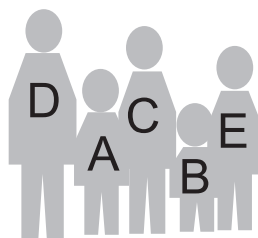
I tabellen ovan visas hur antalet böcker i travarna varierar under 13 dagar. I de kommande testerna, är antalet böcker ≤ 50 , antalet travar som mest ≤ 15 och antalet dagar ≤ 100 .

Körningsexempel (motsvarande tabellen ovan):

Antal travar ? 3
 Böcker i trave 1 ? 4
 Böcker i trave 2 ? 5
 Böcker i trave 3 ? 7

Dag 13 uppkom ett upplägg, som redan förekommit dag 7

UPPGIFT 3 – UPPSTÄLLNING



FIGUR 2. Raden med barn i exemplet sedda bakifrån.

En grupp med n barn, låt oss kalla dem A, B, C och så vidare, beslutar sig för att testa din tankeförmåga. Utan att du ser dem ställer de upp sig på en rad. Sen räknar vart och en av dem hur många av de barn som står till vänster om honom/henne som är längre än han/hon själv, och sedan likadant med dem som står till höger. Var och en skriver ner dessa antal på en lapp som de ger till dig efter att ha frångått uppställningen. Deras enkla uppmaning till dig är att tala om i vilken ordning de stod.

Ett exempel med fem barn visas i figur 2 ovan. A har ett längre barn (D) till vänster om sig och två (C och E) till höger. B har tre längre barn till vänster om sig och ett till höger. C har ett längre barn till vänster om sig men inget till höger och så vidare. Informationen på lapparna kan sammanfattas så här:

Barn	Vänster	Höger
A	1	2
B	3	1
C	1	0
D	0	0
E	2	0

Tyvärr klarade du inte nöten utan måste i hemlighet smyga iväg och skriva ett datorprogram som löser uppgiften. För att underlätta för dig själv nästa gång barnen ansätter dig så vill du kunna variera både antalet barn (mellan 3 och 8, inklusive) och informationen på lapparna. Du kan förutsätta att alla barn är olika långa och att de inte har gjort något misstag när de skrev lapparna. Intressant nog finns det aldrig mer än en lösning. Körningsexempel:

```

Antal barn ? 5
Barn A, vänster ? 1
Barn A, höger ? 2
Barn B, vänster ? 3
Barn B, höger ? 1
Barn C, vänster ? 1
Barn C, höger ? 0
Barn D, vänster ? 0
Barn D, höger ? 0
Barn E, vänster ? 2
Barn E, höger ? 0
    
```

Uppställningen: D A C B E

UPPGIFT 4 – BOKSTAVSLEK

Du får givet en bokstavssekvens med högst 20 bokstäver där alla bokstäver är olika och valda bland versalerna A – Z. Du ska nu genom att utföra *högst* n omflyttningar (där $n \leq 100$) ändra bokstävernas ordning så att den sekvens som bildas kommer så tidigt som möjligt i alfabetisk ordning. Varje omflyttning innebär att du byter plats på två intill varandra stående bokstäver.

Körningsexempel:

Sekvens ? KROATIEN

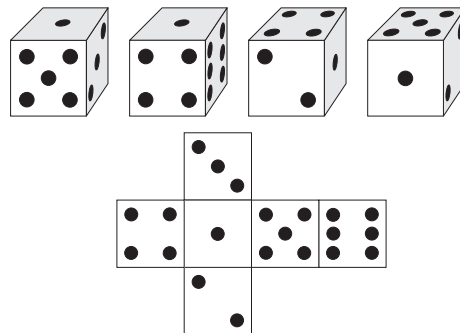
Antal omflyttningar ? 5

Bästa sekvens: AKORITEN

Kommentar: Sekvensen kan till exempel erhållas på följande sätt:

KROATIEN → KORATIEN → KOARTIEN → KAORTIEN → KAORITEN → AKORITEN

UPPGIFT 5 – TÄRNINGAR



FIGUR 3. De fyra tärningarna i exemplet och en möjlighet hur tärning 1, 3 och 4 skulle kunna se ut uppklippt i just det här exemplet. Tärning 2 kan däremot inte ha denna orientering av sidorna.

I den här uppgiften kan sidorna på en tärning vara orienterade hur som helst, det vill säga motstående sidor behöver **inte** nödvändigtvis ha sammanlagt sju prickar och så vidare. Du ska skriva ett program som tar som indata beskrivningar av fyra slagna tärningar. Tre av dessa är egentligen samma tärning men sedd från olika hörn, medan den fjärde inte kan ha samma orientering av sidorna i förhållande till varandra. Programmet ska avgöra vilken av de fyra tärningarna som inte kan vara identisk med de andra.

Programmet ska fråga efter beskrivningen av var och en av de fyra tärningarna. Beskrivningen av en tärning består av tre siffror: antalet prickar på de tre i medurs ordning avlästa sidorna runt ett av tärningens hörn, se figuren 3, skrivna som ett tresiffrigt tal (eller sträng) utan blanksteg. Programmet ska skriva ut numret på den tärning som inte kan vara identisk med de övriga. I varje testfall kommer det att finnas exakt en lösning.

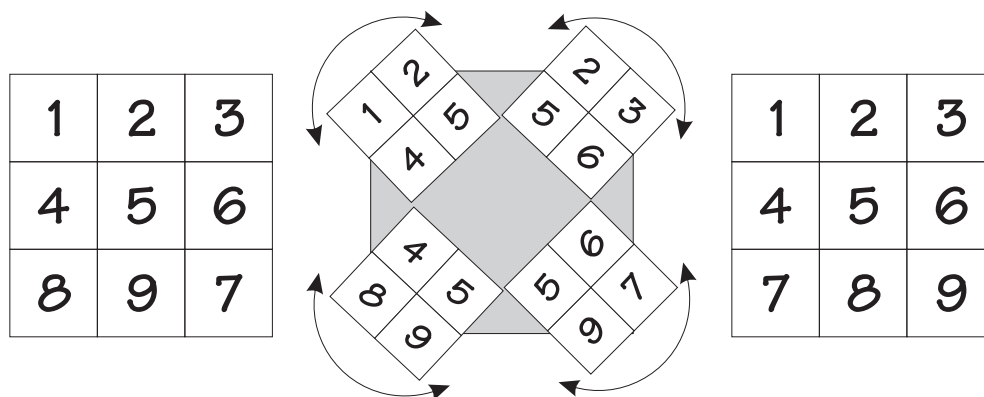
Körningsexempel:

```
Tärning 1 ? 135
Tärning 2 ? 164
Tärning 3 ? 412
Tärning 4 ? 521
```

Udda tärning: 2

Kommentar: Tärningen som beskrivning 1, 3 och 4 visar olika hörn av, kan uppklippt till exempel se ut som i nedre delen av figur 3. Det finns ingen lösning där beskrivning 2 kan visa samma tärning som två av de andra.

UPPGIFT 6 – NOKIASPEL



FIGUR 4. Till vänster ser vi en av många startmöjligheter. I mitten visas hur de fyra 2×2 -kvadraterna kan roteras. Till höger visas målet.

I flera av Nokias mobiltelefoner finns detta förströelsespel eller pussel. Pusslet går ut på att från ett givet utgångsläge, som till exempel till vänster i figur 4, nå fram till ordning och reda som till höger. I varje ställning finns åtta möjliga drag. Varje liten 2×2 -kvadrat kan vridas 90° *medurs* eller *moturs* i taget. Om man skulle göra fyra drag i följd med samma kvadrat i samma riktning skulle man vara tillbaka vid utgångsställningen.

Vi har undersökt att det aldrig behövs fler än 11 drag för att nå målet från vilken utgångsställning som helst. Din uppgift är att skriva ett program som tar emot en ställning och som bestämmer hur många drag den minst kräver för att nå målet.

Ställningen (avläst rad för rad) ges till programmet som en sträng innehållande varje siffra i intervallet $1 \dots 9$ exakt en gång.

Körningsexempel:

Ställning ? 123456897

Denna ställning kräver 6 drag

Bedömning: En effektiv lösning av denna uppgift kan ge tre poäng (övriga uppgifter ger maximalt två poäng). En lösning som klarar alla ställningar vilka kräver högst 8 drag belönas med två poäng.