

Uppgift 1

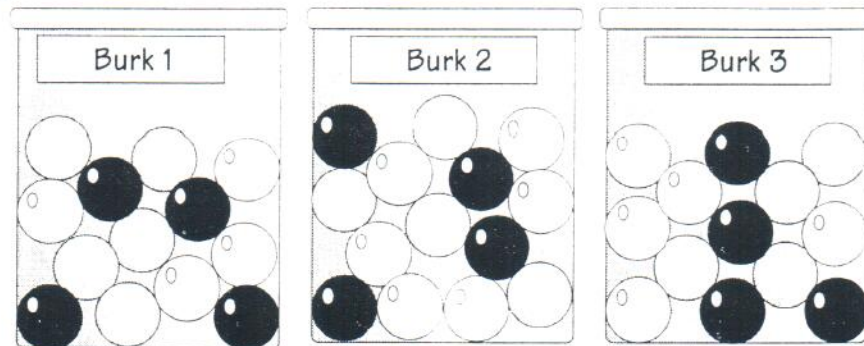


Fig. 1

I de tre burkarna i figuren finns karameller av tre olika sorter (*Gråmunkar*, *Svarta kanonkulor* och *Vita nejlikor*). Den ordningsamme Vilgot vill ordna konfekten så att varje burk innehåller endast en sort. Det spelar ingen roll i vilken burk de olika sorterna hamnar. Det viktiga för Vilgot är istället att han flyttar så få karameller som möjligt.

Skriv ett program som tar emot uppgifter om hur många karameller av varje sort det finns i burkarna och som bestämmer det minsta antalet karameller som måste flyttas för att få godiset uppdelat. Programmet ska också tala om i vilka burkar de olika karamellerna hamnat.

Indata: Programmet frågar genom en rad i taget efter burkarnas innehåll. På varje rad anges i tur och ordning antalet Gråmunkar (G), antalet Svarta kanonkulor (S) och antalet Vita nejlikor (V).

Burk 1 G S V: 5 4 4

Burk 2 G S V: 4 4 6

Burk 3 G S V: 4 4 5

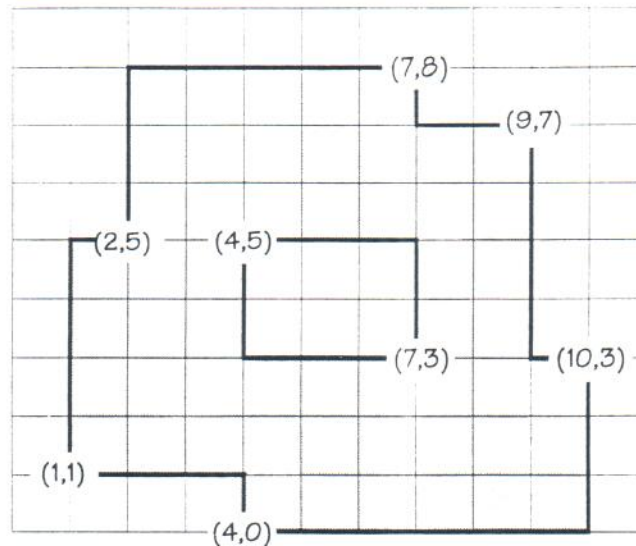
Det totala antalet karameller understiger alltid 30000.

Utdata: Programmet svarar med det antal karameller som har flyttats. Samt en rad med tre bokstäver som anger burkarnas slutliga innehåll, med början för Burk 1.

25 karameller har flyttats.

Burkarnas innehåll: GVS

Uppgift 2



En bonde planerar att inhägnas sina ägor och är intresserad av hur många meter staket som kommer att gå åt till detta. Skriv ett program som tar emot uppgifter om ägornas utseende och som sedan beräknar staketåtgången uttryckt i meter.

Bondens ägor kan beskrivas med hjälp rektanglar. Vi tänker oss en karta, exempelvis som den i figuren. *Denna* kan beskrivas med hjälp av *fyra* rektanglar. Vi tänker oss kartan placerad i första kvadranten av ett koordinatsystem. Genom att ge *nedre vänstra* hörnet och *övre högra* för varje rektangel beskrivs hela egendomen.

En enhet på kartan motsvarar 10 m i verkligheten. Indata ges på ett sådant sätt att ingen överlappning sker. Staketet ska byggas endast mot den mark som han inte äger (motsvarar den tjocka linjen i figuren)

Indata: Först en fråga om antalet rektanglar (≤ 10), som beskriver bondens ägor. Därefter nedre vänstra och övre högra hörnets koordinater för varje rektangel.

Alla koordinater är heltal och tillhör intervallet $[0,50]$

ANTAL ÅKRAR: 4

ÅKER 1: 1 1 4 5

ÅKER 2: 2 5 7 8

ÅKER 3: 7 3 9 7

ÅKER 4: 4 0 10 3

Utdata: En utskrift som anger hur många meter staket som går åt för att inhägnas bondens ägor.

DET GÅR ÅT 440 METER STAKET

Uppgift 3

A	0	L	11	W	22
B	1	M	12	X	23
C	2	N	13	Y	24
D	3	O	14	Z	25
E	4	P	15	Å	26
F	5	Q	16	Ä	27
G	6	R	17	Ö	28
H	7	S	18	.	29
I	8	T	19	?	30
J	9	U	20	└	31
K	10	V	21		

fig 1

```

V Ä L K O M M E N
21 27 11 10 14 12 12 4 13
10101 11011 01011 01010 01110 01100 01100 00100 01101
01011 10110 10110 10100 11100 11000 11000 01000 11011
11 22 22 20 28 24 24 8 27
L W W U Ö Y Y I Ä
    
```

fig 2

Sätten att chiffrera meddelanden är många. Denna uppgift kanske beskriver ett nytt sätt. I fig 1 ser du en tabell över de tecken som kan förekomma i ett meddelande. I kolumnen intill finns ett tal som hör ihop med tecknet. Tecken 31 är mellanslag.

Om vi nu vill chiffrera meddelandet VÄLKOMMEN ersätter vi först de nio bokstäverna med motsvarande tal och vi får den andra raden i figur 2. Efter att ha översatt varje tal i raden till binär form erhålls rad tre som en följd av ettor och nollor. I nästa steg *flyttas det första tecknet* (här en etta) längst bak och vi erhåller den fjärde raden. Genom att dela in denna rad i grupper om fem binära siffror som översätts till decimala tal erhålls rad fem. Återigen använder vi så tabellen och får fram nio nya tecken. Meddelandet VÄLKOMMEN har chiffrerats till LWWUÖYYIÄ.

Skriv nu ett program som tar emot ett **chiffrerat** meddelande och dechiffrerar det till klartext.

Indata: Det chiffrerade meddelandet (på högst 30 tecken) läses in till programmet

MEDDELANDE: LWWUÖYYIÄ

Utdata: Det dechiffrerade meddelandet

MEDDELANDET I KLARTEXT: VÄLKOMMEN

Uppgift 4

Två torn ska byggas av ett givet antal byggklotsar med olika höjd (i cm). Målet är att få de två tornen lika höga, eller så nära detta nu är möjligt.

Skriv ett program som tar emot uppgifter om byggklotsarnas höjd och med hjälp av detta sätter samman dem till två torn vars höjder skiljer sig så litet som möjligt.

Indata: Först en fråga efter antalet byggklotsar (≤ 15) och sedan en fråga om varje byggklots höjd.

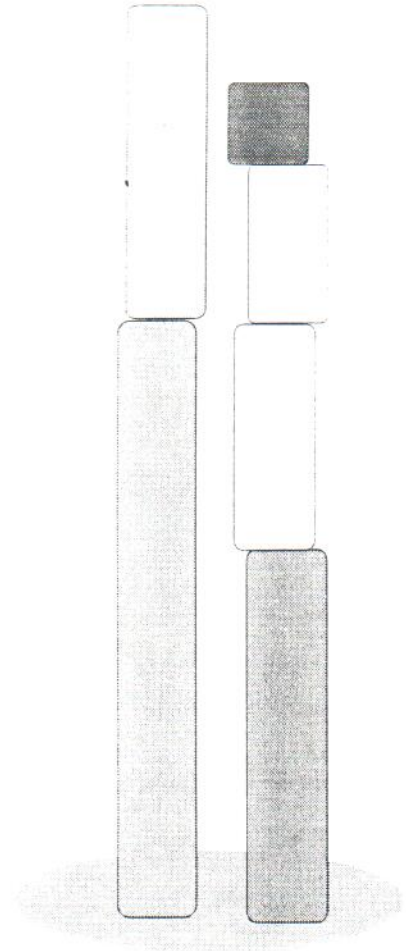
```
ANTAL KLOTSAR: 6
HÖJD FÖR KLOTS 1: 31
HÖJD FÖR KLOTS 2: 54
HÖJD FÖR KLOTS 3: 210
HÖJD FÖR KLOTS 4: 81
HÖJD FÖR KLOTS 5: 109
HÖJD FÖR KLOTS 6: 131
```

Den sammanlagda höjden av samtliga klotsar är alltid mindre än 30000.

Utdata: Det ena tornets höjd följd av en rad med data som anger höjden av de klotsar som ingår i tornet. Två liknande rader med data om det andra tornet.

```
TORN 1 ÄR 319 CM HÖGT
INNEHÅLLER FÖLJANDE KLOTSAR: 109 210
TORN 2 ÄR 297 CM HÖGT
INNEHÅLLER FÖLJANDE KLOTSAR: 31 54 81 131
```

Om det finns mer än en lösning som ger samma resultat räcker det om du presenterar en av dessa.



Uppgift 5



Fig 1. Denna placering av bollarna fungerar inte eftersom bland annat differensen 2 uppkommit två gånger

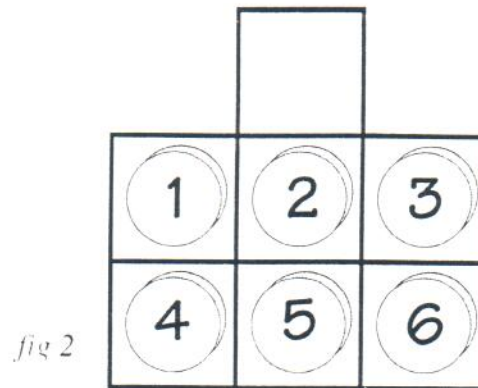
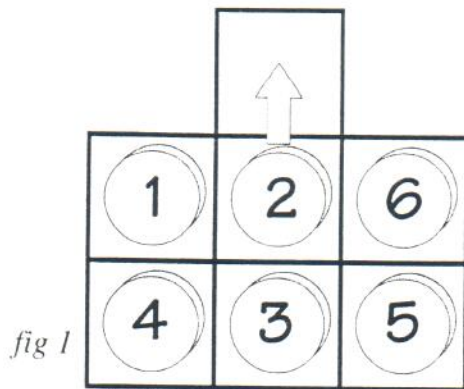
I biljard finns en gren kallad *nummerboll*. Då används 15 bollar numrerade från 1 till 15. Bollarna läggs vid spelets början upp i en triangel som i figuren. Enligt de svenska reglerna har utgångsplaceringen ingen betydelse. Ett av alla möjliga utgångslägen är speciellt intressant. Från numren hos bollarna i översta raden bestäms vilka bollar som ska ligga i andra raden. Genom att bestämma *skillnaden mellan den boll med högsta numret och den med lägsta numret* snett ovanför bestäms vilken boll som ska ligga på aktuell plats. Nu finns det bara en översta rad (och dess spegling) som gör det möjligt att på detta sätt placera ut samtliga 15 bollar.

Skriv ett program som bestämmer upplägget av de 15 bollarna så att regeln, beskriven ovan, gäller för hela upplägget.

Indata: -

Utdata: En lista med fem rader där den första talar om vilka fem bollar som ska ligga i översta raden — i rätt ordning. Andra raden anger sedan i ordning vilka fyra bollar som ska placeras här. Och så vidare. Ingen speciell layout, som ger en triangulär utskrift, är nödvändig.

Uppgift 6



I figur 1 ser vi ett brickspel med sex brickor och en tom plats. Målet med spelet är att utifrån en given startkonfiguration nå fram till målet, som visas i figur 2, genom så få drag som möjligt. I varje drag kan någon av de (högst fyra) brickor, som gränsar till den tomma rutan flyttas dit.

Skriv ett program som tar emot en startkonfiguration och som bestämmer det minsta antalet drag som behövs för att nå målet.

Indata: Indata består av två rader, med vardera tre tal, eftersom det förutsätts att det alltid är topprutan som från början är tom. I dessa rader förekommer förstas talen 1 till 6 precis en gång.

RAD 1: 1 2 6

RAD 2: 4 3 5

Utdata: Först ett tal som anger det minsta antalet drag (≤ 20) för att nå målet och därefter dragen. Man anger ett drag genom att tala om numret på den flyttade brickan.

ANTALET DRAG ÄR 6

DRAGEN: 2 3 5 6 3 2