

UPPGIFT 1

$$\begin{array}{r} \quad \quad \quad *** \\ \quad \quad \times \quad \quad ** \\ \hline ****1 \end{array}$$

1. Två tal ska multipliceras där endast sista siffran i resultatet är given. Trots det finns det bara en lösning då alla siffrorna 0...9 används en gång var

Figuren visar en multiplikation mellan ett tresiffrigt och ett tvåsiffrigt tal. Produkten är femsiffrigt och dess sista siffra är 1. Om man bestämmer att alla de tio siffrorna ska ingå precis en gång i beräkningen finns det bara möjlighet att ersätta asteriskerna – $927 \cdot 63 = 58401$.

Skriv ett program som frågar efter de tre talens *utseende* och som med hjälp detta bestämmer alla lösningar bestående av tre tal, tillsammans innehållande de tio siffrorna 0..9 exakt en gång och där det sista talet är de två första talens produkt. Dessutom ska förstås *de givna siffrorna* behålla sina platser i lösningarna. Observera att inget av de tre talen ska inledas med siffran 0.

Indata: Tre rader som var och en frågar efter ett tal:

```
Faktor1: ***
Faktor2: **
Produkt: ****1
```

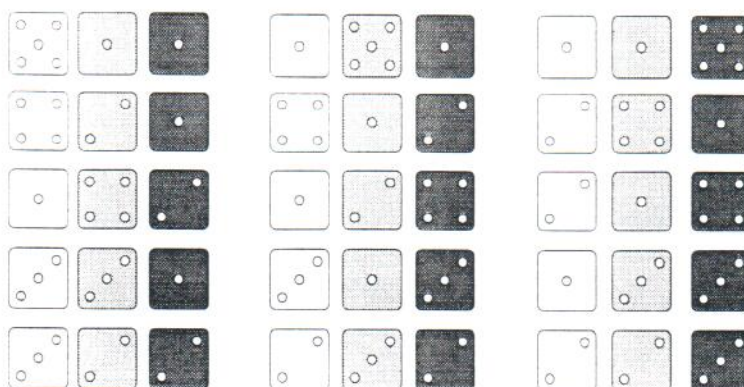
De tre strängarna innehåller förstås alltid tillsammans exakt 10 tecken, alla hämtade bland tecknen *, 1...9.

Utdata: En rad för varje funnen lösning innehållande de tre talen

```
927 63 58401
```

Det kan förutsättas att det finns minst en lösning.

UPPGIFT 2



2. Med 3 olikfärgade tärningar kan summan 7 bildas på 15 olika sätt.

I figuren ser du tre olikfärgade tärningar och alla 15 möjliga kombinationer, att med dessa bilda summan 7.

Skriv ett program som tar emot uppgift om *antal tärningar* och *önskad summa* och som beräknar antalet möjligheter att med dessa tärningar bilda den önskade summan.

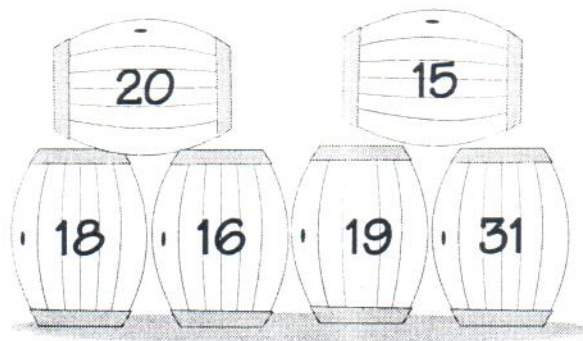
Indata: Programmet ska fråga efter antalet tärningar $1 \leq n \leq 20$ och önskad summa $1 \leq s \leq 120$. Man kan förutsätta att det finns åtminstone en möjlighet att bilda summan med det givna antalet tärningar

Antal tärningar: 3
Önskad summa : 7

Utdata: Utdata består av endast en rad

Summan kan bildas på 15 olika sätt

UPPGIFT 3



3. Vilken av de sex tunnorna innehåller öl?

I figuren ser vi tunnorna i lagret hos en vin och ölförsäljare från den gamla goda tiden. Varje tunna är ordentligt märkt med hur många liter den innehåller. Vi kan dock inte avläsa *vad* den innehåller. Två kunder köpte allt vin försäljaren hade. Den ene köpte dubbelt så mycket som den andre. Återstod endast en tunna – den med öl – vilken var det?

Skriv ett program som tar emot uppgifter om *antal tunnor*, *hur mycket de innehåller* samt *hur många gånger mer vin den ena kunden köpte* (ett heltal). Programmet ska sedan ta reda på hur tunnorna fördelades mellan kunderna och hur mycket den *enda kvarvarande tunnan* (öltunnan) innehåller.

Indata: Uppgifter om antalet tunnor, $3 \leq n \leq 15$, volymen $1 \leq v_i \leq 200$ hos de $i = 1 \dots n$ tunnorna. Dessutom ett heltal $1 \leq x \leq 10$ som anger hur många gånger mer vin den ena kunden köpte.

```
Antal tunnor : 6
Tunnornas volym : 15 16 18 19 20 31
Antal gånger mer: 2
```

Utdata: Resultatet presenteras på tre rader

```
Kund 1: 15 18
Kund 2: 16 19 31
Öltunnan: 20
```

Om det finns flera lösningar räcker det att ange en av dem.

UPPGIFT 4

Genom att sätta in plus- och minustecken på lämpliga ställen i siffersträngen 187234569 kan man bilda aritmetiska uttryck som till exempel

$$\begin{array}{lll} 1 - 8 + 7 - 23 - 45 + 69 = 1 & 1 - 8 - 72 + 3 + 4 + 5 + 69 = 2 & 1 + 8 - 7 - 23 - 45 + 69 = 3 \\ 1 + 8 - 72 - 3 - 4 + 5 + 69 = 4 & 1 - 87 - 23 + 45 + 69 = 5 & 18 + 7 + 2 + 3 + 45 - 69 = 6 \\ 1 - 87 + 23 - 4 + 5 + 69 = 7 & 1 + 8 - 72 + 3 + 4 - 5 + 69 = 8 & 1 - 8 - 7 + 2 - 3 - 45 + 69 = 9 \end{array}$$

Det finns förstås ett minsta tal som *inte* kan bildas på detta sätt, ur den givna siffersträngen. Skriv ett program som tar emot en följd av nio siffror, en av varje 1..9 och som bestämmer det minsta tal som inte kan bildas, genom att placera in plus- och minustecken mellan siffrorna.

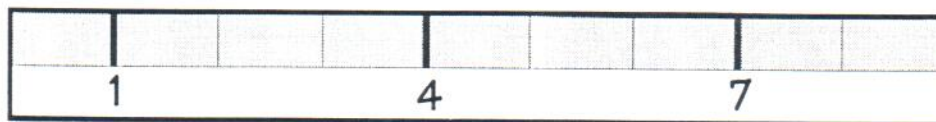
Indata: Indata består av en godtycklig följd av de nio siffrorna 1..9

Sifferföljden: 187234569

Utdata: Endast en rad

Det minsta tal som inte kan bildas är 177

UPPGIFT 5



4. Med denna linjal kan man mäta upp alla längder mellan 1 cm och 9 cm, endast genom att använda linjalens ändar och de tre markeringarna.

I figuren visas en linjal med längden 9 cm. Linjalen har endast tre markeringar vilket är tillräckligt för att det ska gå att mäta upp alla längder i hela cm från 1 cm till 9 cm.

Det går inte att åstadkomma detta med hjälp av färre markeringar. Däremot finns det fyra olika sätt att välja de tre markeringarna.

1	2	6
1	4	7
2	5	8
3	7	8

Skriv ett program som tar emot uppgift om linjalens längd l (i hela cm) och som bestämmer det minsta antal markeringar som behövs för att kunna mäta upp alla mått mellan 1 cm och l cm.

Programmet ska dessutom bestämma antalet olika sätt att placera ut dessa markeringar. Även de lösningar som är speglingar av andra ska räknas..

Indata: Endast en rad, där programmet frågar efter linjalens längd $3 \leq l \leq 23$,

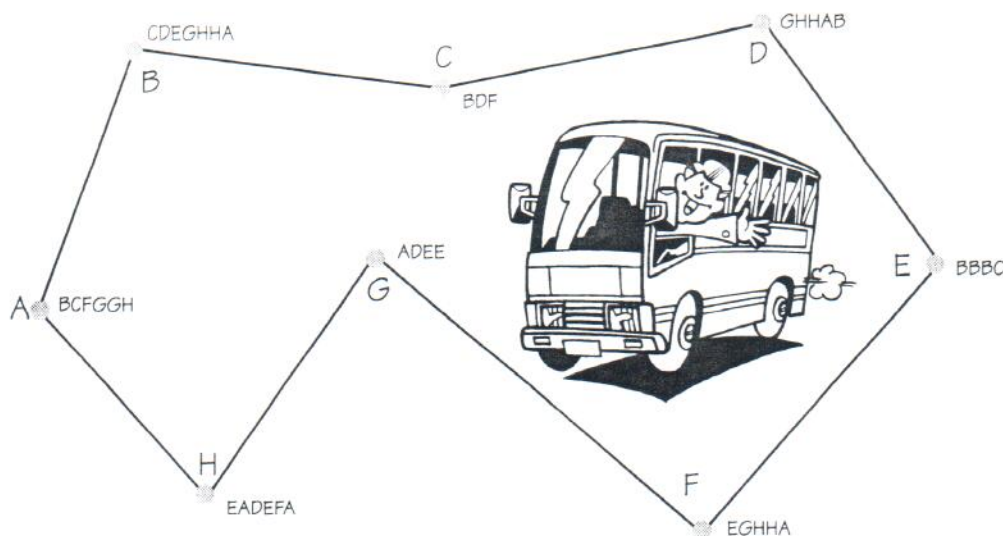
Linjalens längd: 9

Utdata: Utdata består av två rader

Det krävs 3 markeringar

Det finns 4 lösningar

UPPGIFT 6



5. Bussens tur går medurs med start i A. Efter hur många minuter är alla resande hemma?

En ringlinje med 8 hållplatser, A till H, trafikeraras av endast *en* buss. Bussen, som kan ta 10 passagerare, kör turen varv efter varv – medurs – med start i A, där körtiden mellan hållplatserna, i minuter, är följande:

A-B	B-C	C-D	D-E	E-F	F-G	G-H	H-A
3	2	5	4	5	6	5	3

De resande vid hållplatserna står i välordnade köer. När bussen når hållplatsen, där den stannar 1 minut för av- och påstigning, kan antingen kön ta slut eller bussen bli fullsatt.

Skriv ett program som tar reda på hur lång tid det tar från det att bussen öppnar dörrarna för påstigning på hållplats A till dess att den sista passageraren stiger av och alla köer är tomma.

Indata: Programmet ska ta emot namnet på den fil som innehåller *ködata*.

Filnamn: busstur.dat

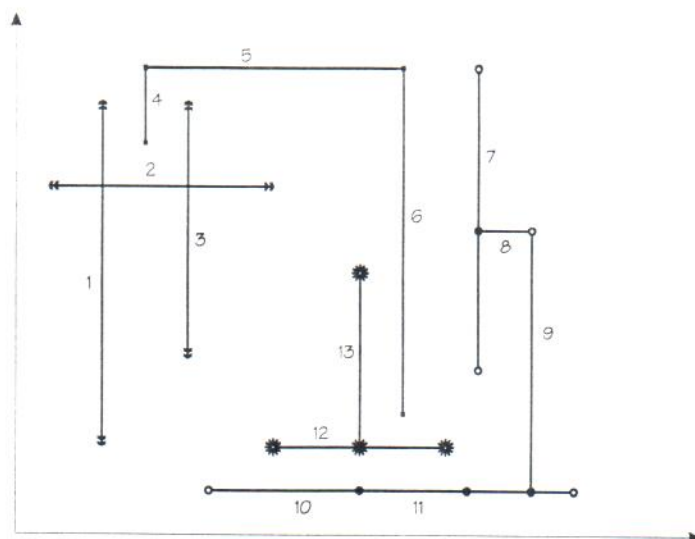
Filen innehåller 8 rader varje rad innehåller mellan 0 och 80 tecken, stora bokstäver A...H. Första raden i filen anger kön för hållplats A och så vidare. Första tecknet på raden anger till vilken hållplats passageraren, som står först i kön, ska resa.

Kön vid en hållplats kan vara tom, vilket medför en tom rad i filen.

Utdata: Utdata består bara av en rad:

Bussturen tog 124 minuter

UPPGIFT 7



6. De 13 sträckorna bildar 4 nät

Figuren visar 13 sträckor. Varje sträcka är antingen *horisontell* eller *vertikal* och alla koordinater är positiva heltal. Det vill säga för sträckans ändpunkter (x_1, y_1) och (x_2, y_2) gäller att antingen $x_1 = x_2$ eller $y_1 = y_2$.

Sträckor som *skär varandra* eller "*snuddar vid varandra*" ingår i *samma nät*. Två sträckor kan även överlappa varandra helt eller delvis (som 11 och 10 i figuren).

Skriv ett program som tar reda på hur många nät det finns och vilka sträckor som tillhör de olika näten.

Indata: Programmet ska fråga efter namnet på den fil som innehåller data om de $1 \leq n \leq 50$ sträckorna.

Filnamn: `strackor.dat`

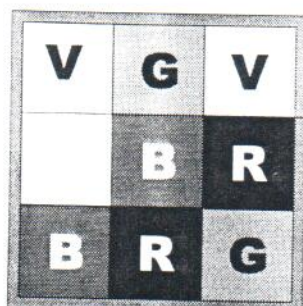
Varje rad på textfilen innehåller data för en sträcka enligt $x_1 \ y_1 \ x_2 \ y_2$ alla heltal, tillhörande $[0, 200]$. Sträckans ordningsnummer motsvarar numret på motsvarande rad i filen.

Utdata: Programmet ska skriva ut varje nät som en parentes med tillhörande sträcknummer:

(1 2 3) (12 13) (5 4 6) (7 8 9 10 11)

Ordningen mellan näten och mellan sträckorna inom parentesen är godtycklig.

UPPGIFT 8



7. Genom 10 drag kan man föra brickan i övre vänstra hörnet till nedre högra

I figuren ses ett pussel av typ 9-spelet, vilket betyder att det har 8 brickor, som om de gränsar till det tomrum, som utgör den nionde kvadraten, kan skjutas dit.

Det finns två brickor av var och en av färgerna Blå, Röd, Gul och Vit, (B, R, G, V), samt en tom plats, (T). Målet med pusslet är att föra den bricka som från början finns i det övre vänstra hörnet till det nedre högra hörnet. *Under tiden får aldrig två brickor med samma färg ligga sida vid sida.*

Skriv ett program som utifrån en given startställning finner *den kortaste* dragserien för att fullfölja uppgiften. Tänk på, att långt ifrån alla utgångsställningar, har någon lösning. Ingen av lösningarna kräver mer än 24 drag.

Indata: Genom en rad ges utgångsställningen:

Startställning: VGVTRBRG

Brickorna ges radvis med början från översta raden. Inte ens från start är det tillåtet att två intilliggande brickor har samma färg.

Utdata: Resultatet skrivs ut genom en följd av bokstäver motsvarande färgen på den bricka som ska flyttas:

Dragserien: V G B V B R V R G V

Om det finns mer än en lösning med samma antal drag räcker det att ange en av dem.