

# Uppgift 1

Skriv ett program som bestämmer det minsta positiva heltal  $n$ , som går att skriva som summan av två kvadrater på fyra olika sätt.

$$n = a^2 + b^2 = c^2 + d^2 = e^2 + f^2 = g^2 + h^2$$

där  $a, b, c, d, e, f, g$  och  $h$  alla är olika positiva heltal.

*Indata:* -

*Utdata:* Det eftersökta talet 1105

# Uppgift 2

En springare startar i hörnet  $a1$ , på ett vanligt  $8 \times 8$  schackbräde. Ett drag väljs slumpmässigt och med samma sannolikhet, ur mängden av tillåtna drag. Ett tillåtet drag är ett drag som görs enligt reglerna för springardrag i schack, se diagram, och till en ruta som tidigare inte varit besökt.

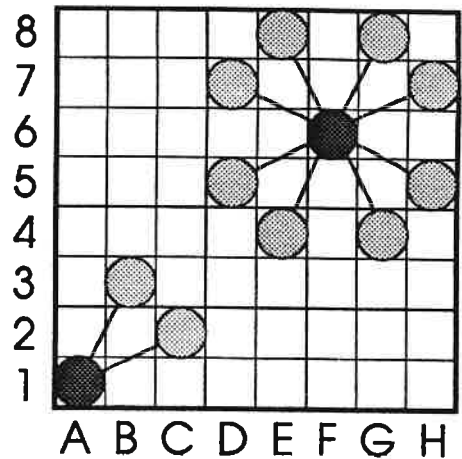
Skriv ett program som genom upprepade försök bestämmer ett medelvärde för denna slumpmässiga

draglängd. draglängd draglängd

OBS Utgångsrutan,  $a1$ , anses inte använd och kan alltså beträddas senare.

*Indata:* Antal försök

*Utdata:* Det beräknade medelvärdet



Springaren på  $a1$  har 8 möjliga drag, till  $b3$  eller  $c2$ . Springaren på  $f6$  kan flyttas till maximala åtta olika rutor:  $e4, g4, d5, h5, d7, h7, e8, g8$

1000 : 33,27

## Uppgift 3.

Rutmönstret intill är en karta över en skärgård. De grå rutorna symboliserar land och de vita, vatten. Två grå, ortogonalt anslutna, rutor är land på samma ö, se rutorna A och B. Däremot finns ingen landförbindelse diagonalt, se rutorna C och D.

Skriv ett program som tar emot en kartbeskrivning och härur bestämmer antalet öar och dess landarea samt den totala vattenarean. Arean räknas i rutor. Kartan består alltid av 8x8 rutor.

*Indata:* Kartan på formen

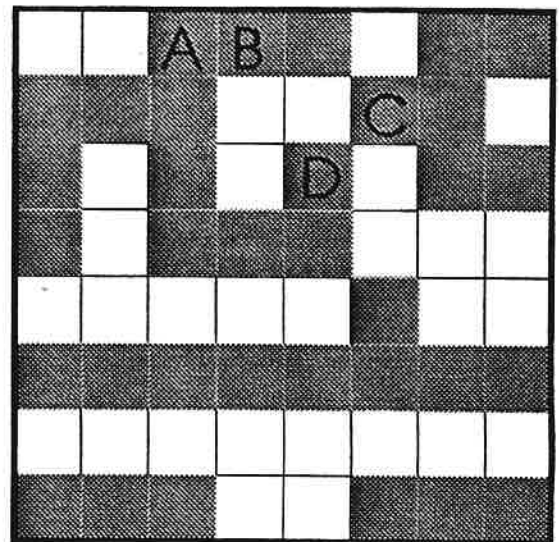
```
  * * *  * *
* * *    * *
*  *  *  * *
*  * * *
      *
* * * * * * *
* * *    * * *
```

För vatten används mellanslag och för land asterisk. De åtta raderna ska inte behöva vara exakt 8 tecken. För en rad utan asterisker ska man till exempel kunna trycka return direkt.

*Utdata:*

```
Ö 1 har arean 13
Ö 2 har arean 6
Ö 3 har arean 9
Ö 4 har arean 3
Ö 5 har arean 3
Total vattenarea 30
```

Ordningen i listan är godtycklig.



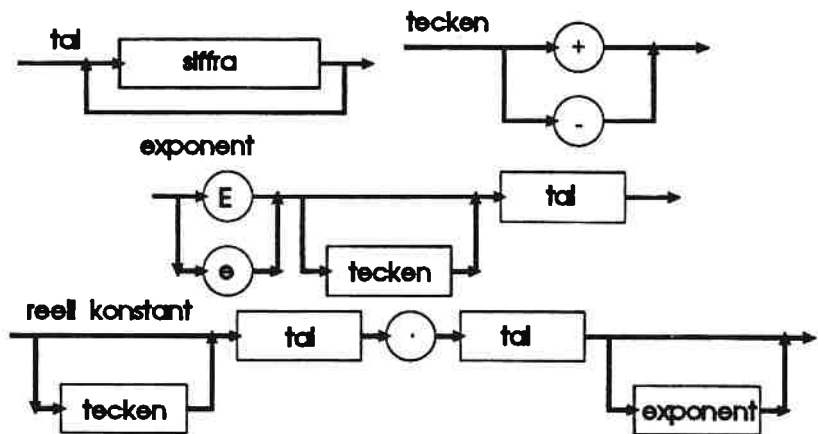
## Uppgift 4

Skriv ett program som avgör om en sträng, på maximalt 80 tecken, är en godkänd *reell konstant* enligt reglerna nedan.

OBS Då definitionen ovan inte helt överensstämmer med den för TurboPascal kan inte inbyggda funktioner användas för att lösa problemet.

*Indata:* En sträng på högst 80 tecken.

*Utdata:* JA, om strängen motsvarar en reell konstant. NEJ om strängen avviker från reglerna för hur en reell konstant får se ut.



-1234.0e+57

## Uppgift 5

Vårt myntsysteem innehåller som bekant 10 öre, 50 öre, 100 öre och 500 öre. Ibland när man står och plockar med sina tioöringar undrar man om valörerna på våra mynt verkligen är de rätta! Nedan ska beskrivas ett mått som försöker mäta effektiviteten hos en uppsättning myntvalörer.

$$E(S) = \frac{1}{A(S) \cdot N(S)}$$

$E(S)$  är ett effektivitetsmått för  $S$ , en given mängd valörer.

$N(S)$  är antalet valörer i  $S$ .

$A(S)$  är ett medelvärde av det minsta antalet mynt som krävs för varje belopp från 10 öre till 1000 öre.

Att bestämma  $E(S)$ , då till exempel  $S=\{10, 50, 100, 500\}$  handlar till största delen om att bestämma det minsta antalet mynt som behövs för att framställa samtliga belopp, 10 öre, 20 öre, ... 990 öre, 1000 öre. Den totala antalet av dessa mynt divideras sedan med 100, antalet olika belopp, för att erhålla  $A(S)$ . I vårt exempel blir  $A(S)=1/100(1+2+3+4+1+2+\dots+2)=5.02$ . Då  $N(S)=4$  betyder det att för vårt nuvarande myntsysteem är  $E(S)=0.0498$

Skriv ett program som beräknar  $E(S)$  för ett givet myntsysteem. Viktigt för myntsysteem är förstås att alla belopp kan erhållas, vilket betyder att tioöringen alltid måste finnas med i  $S$ .

*Indata:* Antalet valörer i systemet, programmet ska klara upp till 7 valörer. De olika myntens valör. En valör ska vara en multipel av 10 som till exempel 130-öringen eller 370-öringen.

*Utdata:*  $E(S)$  för givna indata.

## Uppgift 6

I *fig 1* visas ett enkelt solitärspel med sju rutor och sex brickor, tre svarta och tre vita. Uppgiften består i att med hjälp av ett antal drag byta plats på de svarta och de vita brickorna, *fig 2*.

*Figur 2* visar fyra tillåtna drag i en viss situation. Man får alltså flytta en bricka till den intilliggande rutan eller hoppa över en bricka och hamna på den tomma rutan. Man behöver inte alternera brickornas färg mellan dragen, utan kan till exempel flytta två vita brickor i följd.

Skriv ett program som listar den unika dragföljden som når målet efter 15 drag då första draget görs med en svart bricka.

*Indata:* -

*Utdata:* Den eftersökta dragföljden på formen 3-4, 5-3, ... där siffrorna anger frånruta och tillruta enligt numrering i figuren.

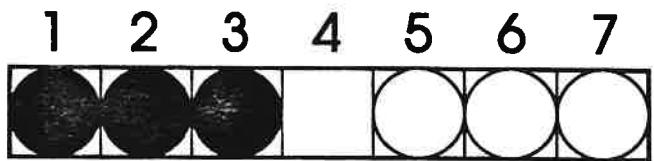


fig 1



fig 2

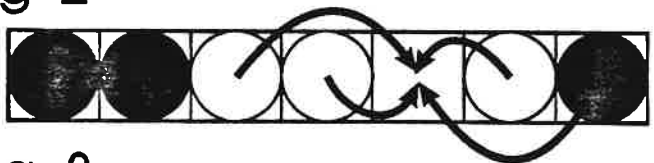


fig 3

*Fig 1* visar utgångsställningen och *fig 2* slutställningen. I ställningen i *fig 3* finns fyra olika drag: 3-5, 4-5, 6-5 och 7-5. Brickorna kan alltså flyttas i båda riktningarna oavsett färg.