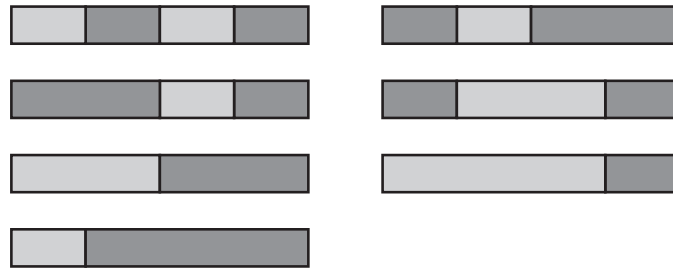


UPPGIFT 1 – PLANKAN



FIGUR 1.

Man vill skapa en längre *planka* med hjälp av ett antal mindre *brädor*. Det finns tre olika typer av brädor, som har längden 1, 2 respektive 3 meter. Det finns ett obegränsat antal av varje typ.

Skriv ett program som bestämmer på hur många olika sätt man kan åstadkomma en plankan av längden n , $1 \leq n \leq 24$.

I figuren ser du de 7 olika sätten att skapa en plankan med längden 4 meter. Dialogen från en programkörning:

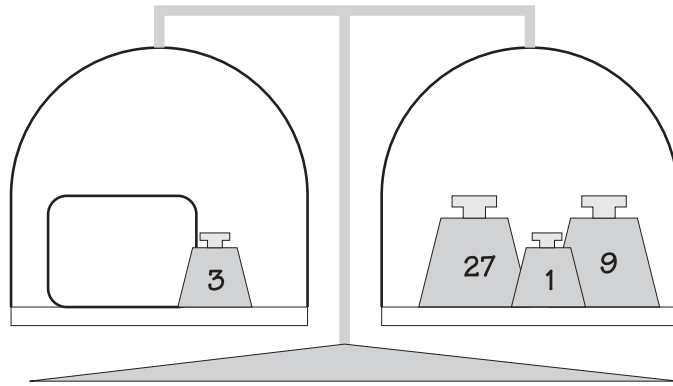
Indata: Plankans längd efterfrågas. Ett heltal i intervallet $1 \dots 24$

Plankans längd: 4

Utdata: Antalet sätt som plankan kan sättas ihop av de mindre brädstumparna.

Det finns 7 olika sätt

UPPGIFT 2 – VÅGEN



FIGUR 2.

Med hjälp av kilovikterna $1, 3, 9, 27 \dots 3^n$ (en av varje) och en *balansvåg* kan man ta reda på vikten hos alla föremål som väger ett helt antal kilo, bara n är tillräckligt stort. I figuren ser vi hur paketet med vikten 34 kg balanseras av de fyra vikterna 1, 3, 9 och 27. Detta är det enda sätt vägningen kan göras på. Det finns alltid, precis ett sätt att utföra vägningen, oavsett paketets vikt. Skriv ett program som tar emot paketets vikt v , och som bestämmer vilka vikter som ska placeras i de två vågskålarna. *Paketet antas alltid placeras i vänster vågskål.*

Indata: Programmet tar emot paketets vikt v , $v \leq 10000$

Vad väger paketet? 34

Nödvändiga vikter, alla från följden ovan, finns tillgängliga.

Utdata: Programmet skriver ut en lista över de vikter som ska användas och i vilken vågskål de ska placeras.

```
Vikt 1 i högra  
Vikt 3 i vänstra  
Vikt 9 i högra  
Vikt 27 i högra
```

Det vikter som inte används ska förstås heller inte vara med i utskriften.

UPPGIFT 3 – SIFFERSUMMA

971 $\rightarrow 9^2 + 7^2 + 1^2 = 131$	863 $\rightarrow 8^2 + 6^2 + 3^2 = 109$
131 $\rightarrow 1^2 + 3^2 + 1^2 = 11$	109 $\rightarrow 1^2 + 0^2 + 9^2 = 82$
11 $\rightarrow 1^2 + 1^2 = 2$	82 $\rightarrow 8^2 + 2^2 = 68$
2 $\rightarrow 2^2 = 4$	68 $\rightarrow 6^2 + 8^2 = 100$
4 $\rightarrow 4^2 = 16$	100 $\rightarrow 1^2 + 0^2 + 0^2 = 1$
16 $\rightarrow 1^2 + 6^2 = 37$	
37 $\rightarrow 3^2 + 7^2 = 58$	
58 $\rightarrow 5^2 + 8^2 = 89$	
89 $\rightarrow 8^2 + 9^2 = 145$	
145 $\rightarrow 1^2 + 4^2 + 5^2 = 42$	
42 $\rightarrow 4^2 + 2^2 = 20$	
20 $\rightarrow 2^2 + 0^2 = 4$	

I den vänstra delen av tabellen ovan startar vi med talet 971. Genom att beräkna summan av *kvadraterna på talets ingående siffror* får vi genom $9^2 + 7^2 + 1^2 = 131$. Går vi sedan vidare med talet 131 på samma sätt, kommer vi via talen 11 och 2 till talet 4. När vi sedan fortsätter kommer vi över 16, 37, 58, 89, 145, 42, 20 åter till 4 – vi har hamnat i en loop!

I tabellens högra del däremot startar vi med talet 863 och når via talen 109, 82, 68 och 100 till talet 1. Startar vi med n , $1 \leq n \leq 1000$ kommer vi antingen att komma fram till talet 1 eller att hamna i en loop.

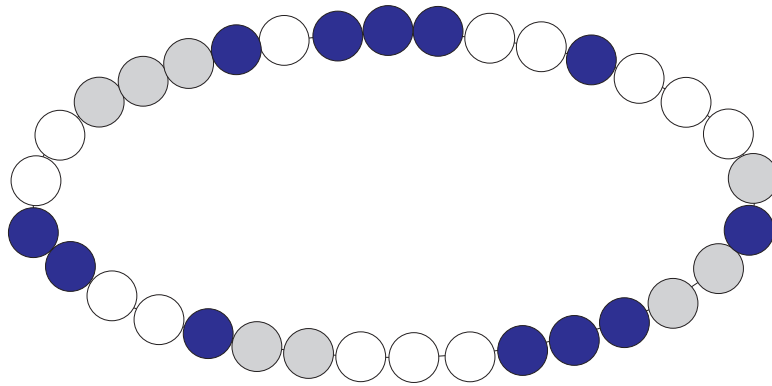
Skriv ett program som bestämmer för hur många av talen 1...1000 man når 1.

Indata: –

Utdata: Bara en enda rad som ger svaret:

Antalet tal, där följderna avslutas med 1 : ???

UPPGIFT 4 – HALSBANDET



FIGUR 3.

I figuren visas ett halsband med pärlor i tre olika färger: *svarta* (S), *grå* (G) och *vita* (V). Halsbandet kan beskrivas som en följd av tecken (≤ 80):

GGGSVSSSVVSVVVGSGGSSSVVVGGSVVSSVV

Det finns förstås många sätt att beskriva samma halsband, beroende på var man sätter början. Man vill nu, genom att plocka bort några pärlor, få så många pärlor som möjligt med samma färg i en obruten följd i halsbandet. De pärlor man plockar bort måste dock alla *ligga intill varandra* (i sin tur vara en obruten följd) men kan vara av vilken färg som helst.

Skriv ett program som tar emot uppgift om halsbandets utseende och som bestämmer hur lång obruten följd av *likfärgade* pärlor, som kan åstadkommas genom att plocka bort en eller flera pärlor i en obruten följd.

Indata:

Halsbandet: GGGSVSSSVVSVVVGSGGSSSVVVGGSVVSSVV

Utdata: Ett körningsexempel (enligt figuren):

Man kan få 6 V pärlor i rad genom att ta bort 7 pärlor

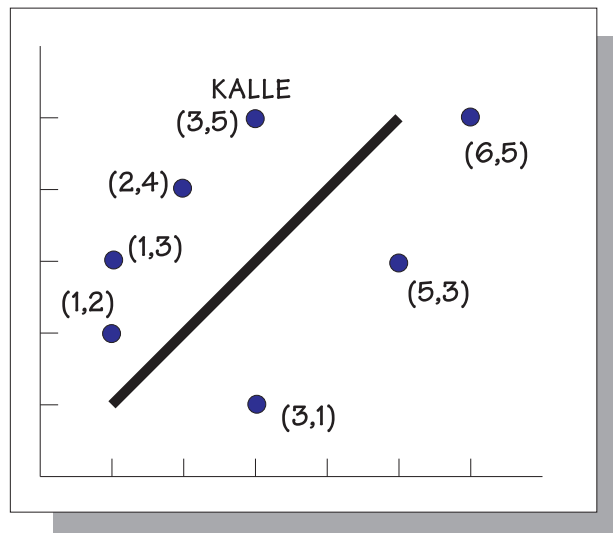
Om samma mål kan nås på flera sätt, så gäller det fall, *där man tar bort så få pärlor som möjligt*. Närliggande svar som inte riktigt räcker till från exemplet:

Man kan få 6 S pärlor i rad genom att ta bort 10 pärlor

Man kan få 6 S pärlor i rad genom att ta bort 17 pärlor

Man kan få 6 V pärlor i rad genom att ta bort 20 pärlor

UPPGIFT 5 – PLANKET



FIGUR 4.

På skolgården finns ett *plank*, så högt att ingen av skolans elever kan se över det. Du ska skriva ett program som avgör vilka av sina kamrater Kalle kan se.

I figuren ser du ett exempel på en karta över plank och skolgård. Koordinater i uppgiften är alltid heltal i intervallet $0 \dots 30$. Plankets ändrar finns i koordinaterna (1, 1) och (5, 5) och Kalle finns i (3, 5). Därifrån kan ha se kamraterna i (1, 2), (1, 3), (2, 4) och (6, 5). Av detta framgår att en kamrat inte helt kan skymma en annan ((2, 4) skymmer inte (1, 3)) samt att de som står i linje med plankets ändpunkter och Kalle också syns (här (6, 5)). Inga barn befinner sig i de koordinater som planket passerar.

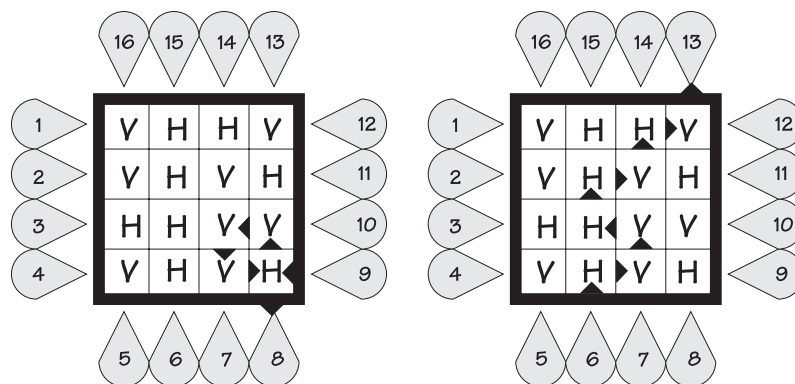
Indata: Först plankets ändpunkter, sedan n , $n \leq 10$, som anger antalet barn på skolgården och därefter n koordinatangivelser, en för varje barn. Den första av dessa koordinater anger var Kalle finns. Exemplet från figuren ger:

```
Ena änden av planket: 1 1
Andra änden av planket: 5 5
Antal barn: 7
Barn 1: 3 5
Barn 2: 5 3
Barn 3: 6 5
Barn 4: 2 4
Barn 5: 1 2
Barn 6: 3 1
Barn 7: 1 3
```

Utdata: En lista av koordinater som anger vilka barn Kalle kan se:

```
Kalle kan se: (6,5) (2,4) (1,2) (1,3)
```

UPPGIFT 6 – ROBOT



FIGUR 5. Till vänster ser vi ingång genom dörr 9 och utgång genom dörr 8. Till höger sker entrén genom dörr 6 och sortin genom 13.

Ett kvadratisk rum för en rörlig men ganska korkad robot är indelad i 4×4 rutor. Under golvet i varje ruta sitter en sändare som ger en instruktion till roboten, när den befinner sig på den rutan. En sändare ger alltid samma instruktion. Roboten förstår två instruktioner:

- H höger, det vill säga, roboten ska vrida sig medurs och sedan gå en ruta.
- V vänster, det vill säga, roboten ska vrida sig moturs och sedan gå en ruta.

Notera att höger och vänster är relativt det senaste gångsteget.

Till gården hör 16 dörrar numrerade enligt figuren

Om man släpper in roboten genom en viss dörr kommer den att gå runt enligt instruktionerna tills den slutligen kommer ut genom en annan (eller möjligen samma) dörr. Din uppgift är att skriva ett program som listar ut vilken instruktion som ges i varje ruta där indata till programmet är, var roboten kommer ut för varje dörr den släpps in genom. Till de indata som ges finns alltid minst en lösning. Om det finns flera lösningar räcker det att ange en.

Indata: 16 heltal. Det första anger numret på den dörr där roboten kommer ut om den släpps in genom dörr 1, och så vidare.

Utgångsdörrar: 16 1 6 5 4 13 10 9 8 7 14 11 12 15 2 3

Utdata: En *karta* över gården där varje ruta är angiven med den instruktion som ges där.

VHHV
 VHVH
 HHVV
 VHVH