

Uppgift 1

Platserna i biografsalongen är numrerade. Varje rad består av ett jämnt antal stolar. Den ena halvan av en sådan rad fylls på av biobesökare som släpps in från vänster och den andra halvan av dem som släpps in från höger. Besökarna anländer slumpmässigt och söker direkt upp sina platser. När någon, som ska sitta längre in mot mitten anländer, måste någon eller några ibland resa sig för att ge fri passage. I kväll är det utsålt!

Nedan visas en halv stolsrad. Talen anger hur många gånger biobesökaren på just den platsen blev tvungen att resa sig innan raden var fullsatt. Platsnummer räknas från vänster till höger.

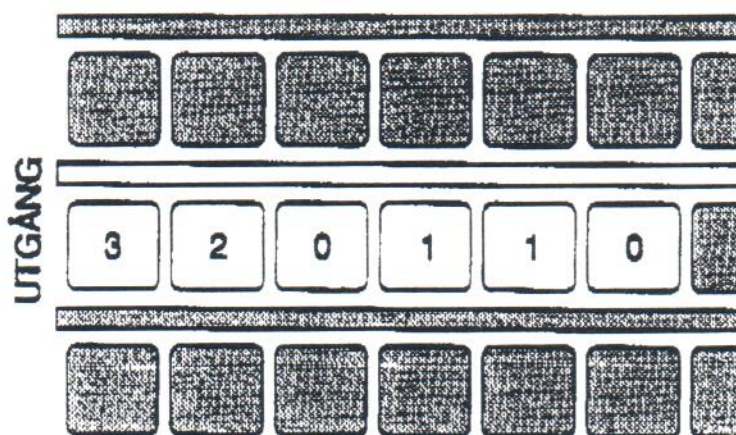


fig 1

Skriv nu ett program som tar emot uppgifter om hur många gånger varje besökare, på en halv stolsrad, fick lov att resa sig och som därefter beräknar och skriver ut i vilken ordning personerna anlände.

Indata: Uppgift om antalet platser i den halva stolsraden, maximerat till 20. Resningsfrekvensen för varje plats med början vid platsen närmast utgången. Programmet kan förutsätta att indata är korrekta, det vill säga att det alltid finns en lösning.

Utdata: Platsnumren i ordning från den som anlände först till den som anlände sist.

Uppgift 2

I denna uppgift ska du ta reda på om den *svarta kungen* står i schack för den *vita löparen*, det *vita tornet* eller möjligtvis båda. På brädet finns alltid, förutom dessa tre pjäser, också en *svart bonde*. I de två diagrammen nedan visas löparens och tornets gång i schack.

- ☐ Den vita löparen rör sig diagonalt, men stoppas upp av annan pjäs.
- ☐ Det svarta tornet rör sig ortogonalt, men stoppas upp av annan pjäs.

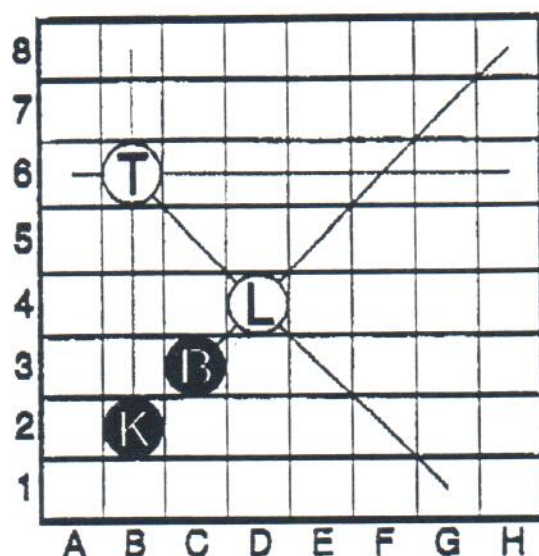
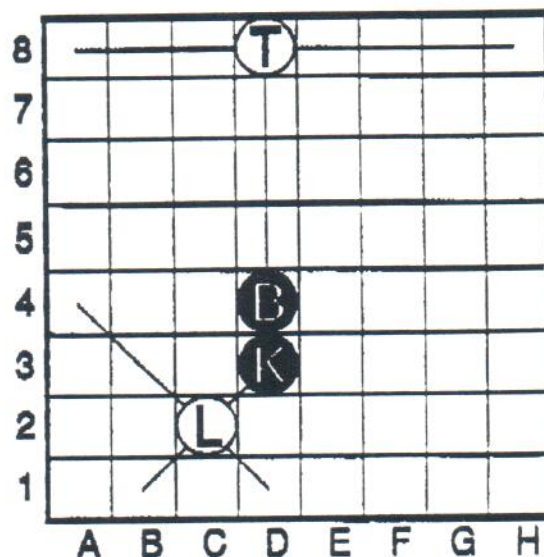


fig 2



I det vänstra diagrammet schackar det vita tornet svart kung. Den svarta bonden hindrar däremot den vita löparen från att schacka svart kung. I det högra diagrammet schackar den vita löparen svart kung medan den svarta bonden står i vägen för en schack från det vita tornet.

Indata: Programmet frågar efter de fyra pjäsernas placering. Indata ges i traditionell schacknotation, exempel A1, B4, H7 vars betydelse framgår av diagrammen ovan. För enkelhet skull använder vi bara versaler, stora bokstäver och förutsätter att indata är korrekt.

Utdata: Oftast en men någon gång två av följande tre rader.

- ☐ Den vita löparen schackar svart kung
- ☐ Det vita tornet schackar svart kung
- ☐ Ingen pjäs schackar svart kung

Uppgift 3

Under 1970-talet lanserades ett tankespel vid namn Master Mind. Här följer spelreglerna.

Spelet har två deltagare. En *kodhållare* och en *kodknäckare*. Kodhållaren bestämmer sig i hemlighet för en *fyrbokstavig kod* där där varje bokstav är någon av A, B, C, D, E eller F. Alla kombinationer av dessa är tillåtna. (I Master Mind väljer i verkligheten kodhållaren fyra plastknoppar från sex olika färger.)

Nu är det kodknäckarens sak att med så få gissningar som möjligt komma på kodhållarens hemliga kod. Varje gissning består av en fyrbokstavig kod. Kodhållaren talar om hur gissningen gick genom att meddela: hur många bokstäver som är *rätt placerade* och hur många som finns med i den hemliga koden men *på fel plats*. Om exempelvis den hemliga koden är ABBD och gissningen BDBF, så finns *en* av den första kategorin, position 3, fortsättningsvis markerad med +, och *två* av den andra, fortsättningsvis markerad med -. Nedan följer två fullständiga spel:

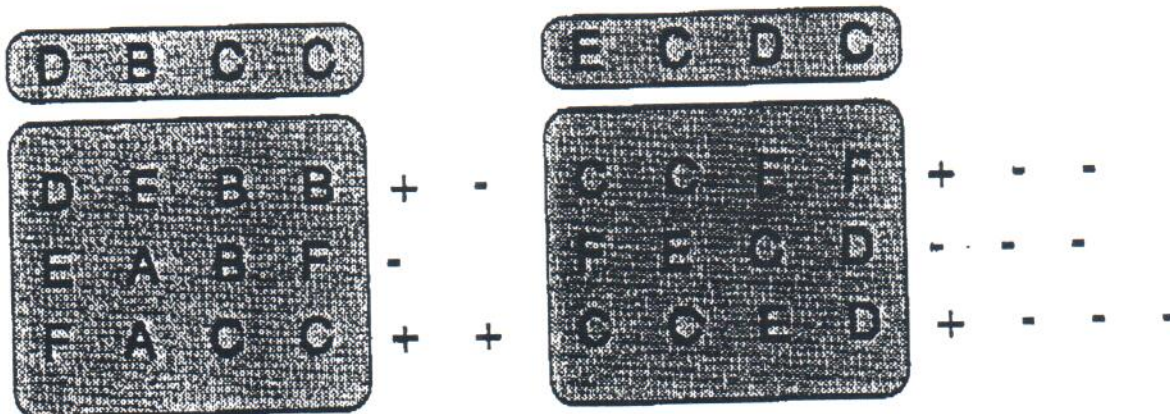


fig 3

Överst, fig 3, visas den hemliga koden och därunder kodknäckarens tre gissningar med resultat. I båda dessa exempel finns bara en *bra* fjärde gissning, som då alltså är den hemliga koden. Observera att varje bokstav i den hemliga koden och i gissningen högst kan ingå i en relation och att relationer av typ *rätt plats* har högst prioritet, se fig 4.

Uppgiften går nu ut på, att med utgångspunkt från ett antal gissningar och svar till dessa, låta ett program lista samtliga möjliga hemliga koder, fyrbokstaviga strängar, som kan vara en lösning. Programmet ska alltså finna samtliga förslag till hemliga koder som ger samma svar för koderna i indata.

Indata: Exempel på dialog

```
Antal gissningar: 2
Gissning   : ABDD
Rätt       : 2
Fel plats  : 1
Gissning   : BFFA
Rätt       : 0
Fel plats  : 1
```

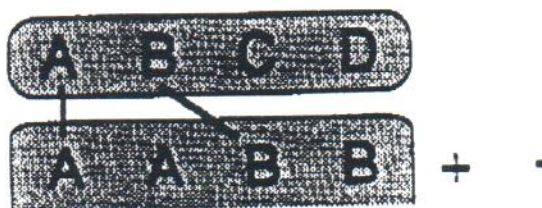


Fig 4.

Indata kan antas vara korrekt. Gissning består av fyra stora bokstäver mellan A och F. Rätt (+) och fel plats (-) är ett heltal i intervallet [0..4]. Däremot är det inte sagt att det alltid måste finnas en lösning.

Utdata: En lista över alla koder som kan stämma in på givna indata. Om ingen sådan kod kan återfinnas ska detta meddelas.

Uppgift 4

Skriv ett program som tar emot ett uttryck av följande utseende

$$(x-4)(x+2)(x+1)(x-3)(x+1)$$

och som sedan expanderar uttrycket och skriver ut resultatet

$$x^5-3x^4-11x^3+15x^2+46x+24$$

Uttrycket ska bestå av mellan 1 och 9 parenteser där varje parentes består av *fem* tecken enligt följande

- ☐ vänsterparentes
- ☐ bokstaven x
- ☐ + eller -
- ☐ ett tal i intervallet [0..9]
- ☐ högerparentes

Parenteserna fogas sedan samman som i uttrycket ovan utan hjälp av multiplikationsoperator.

Indata: En sträng på maximalt 45 tecken enligt reglerna ovan.

Utdata: För enkelhet skull, en tabell över polynomets termer där gradtal och koefficient anges.

Uppgift 5

Skriv ett program som fördelar de n första positiva heltalen på tre mängder så att om talen a och b finns i samma mängd så får inte $a+b$ finnas i samma mängd.

- ☐ talet 13 kan *inte* ingå i mängden [1,2,4,9], ty $4+9=13$
- ☐ talet 15 kan ingå i mängden [6,7,10,11,12] eftersom denna summa inte kan bildas med hjälp av två medlemmar

För små värden på n är det enkelt att finna en lösning. För större värden finns kanske ingen. Det största n för vilket det finns en lösning är mindre än 30. Tänk på att det troligen kommer att ta betydligt längre tid för programmet att visa att lösning saknas än att finna en av många.

Indata: Programmet frågar efter heltalet n

Utdata: En lista över de tre mängdernas innehåll för en av många lösningar. Om lösning saknas ska detta anges.

Uppgift 6

$$257 + 491 + 863 = 1611$$

Summan ovan består av tre tresiffriga tal där varje tal är ett primtal, (ett primtal p är ett positivt heltal som inte har några andra delare än 1 och p). Dessutom används siffrorna 1 till 9 endast en gång. Man kan bilda mer än 100 sådana summor.

Skriv ett program som bestämmer den största respektive den minsta av dessa summor. Om flera summor ger samma extremvärde ska alla redovisas.

Indata: -

Utdata: Största summan och de tre termerna i samtliga summor som ger denna summa. Minsta summan och de tre termerna i samtliga summor som ger denna summa.