

Programmeringsolympiaden 2009

Kvalificering

TÄVLINGSREGLER

- Tävlingen äger rum på ett av skolan bestämt datum under sex timmar effektiv tid.
- Tävlingen består av sex uppgifter som samtliga ska lösas genom ett datorprogram.
- Uppgifterna ska lösas i valfritt programmeringsspråk. Du får till och med byta språk mellan olika uppgifter.
- Dina lösningar kommer att testköras med förpreparerade indata. Klarar ditt program alla testerna får du 2 poäng för uppgiften. Delpoäng (1 poäng) kan komma att utdelas då programmet inte klarar alla testerna. Ingen närmare bedömning av programkoden görs.
- Samtliga uppgifter leder fram till program vars exekveringstid normalt bör understiga 1 sekund. Skulle en lösning leda fram till ett program vars exekveringstid överstiger 10 sekunder bedöms programmet för det testexemplet som felaktigt.
- Rättningen utförs på samma eller likvärdig dator. Om, vid rättningen, fel uppstår vid kompilering bedöms programmet direkt som felaktigt och lösningen ger 0 poäng.
- Ingen test av indata behöver göras. Alla testdata följer de specifikationer som givits i uppgiften. Om det trots detta, vid rättningen, uppstår exekveringsfel vid körning av programmet bedöms programmet som felaktigt för det testexemplet.
- Deltagandet är individuellt vilket bland annat innebär att inget utbyte av idéer eller filer får ske under tävlingen.
- Internet får inte användas under tävlingen.
- Hjälpmedel: Programspråkens manualer, minst ett exemplar tillgängligt i datorsalen. Formelsamling och räknedosa för varje deltagare.
- Tävlingsbidraget ska lämnas in i form av källkodsfiler som läggs i roten på utdelad diskett eller i en av läraren angiven hårddiskcatalog. Filerna ska döpas till uppg1...uppg6 med passande filtillägg. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

Årets International Olympiad in Informatics anordnas i Bulgarien i augusti. Kanske blir du en av dem som representerar Sverige där.

Lycka till!

UPPGIFT 1 – LAMPOR

Användningen av glödlampor är på väg att avskaffas inom EU, men redan nu kan det löna sig att använda lågenergilampor. Du ska skriva ett program som, givet hur mycket en lampa är tänd varje dag samt elpriset, beräknar efter hur många dagar totalkostnaden (inköp av lampor + elkostnad) för första gången är lägre för lågenergilampan än för glödlampan.

Vi antar följande data:

	Glödlampa	Lågenergilampa
Effekt (W)	60	11
Livslängd (timmar)	1000	8000
Inköpspris (kronor)	5	60

För enkelhets skull antar vi att alla exemplar har exakt den givna livslängden. Detta innebär alltså att för att ha lampan tänd i 1000 timmar räcker det att köpa en glödlampa, men för att ha den tänd i 1001 timmar måste man köpa ytterligare en glödlampa eftersom den första går sönder efter 1000 timmar. Elkostnaden K för att ha lampan tänd i H timmar kan beräknas med formeln

$$K = E \cdot H \cdot P / 100000$$

där E är lampans effekt i W och P är elpriset i öre/kWh. I samtliga testfall kommer endast en lågenergilampa att behöva köpas (lågenergi blir alltså billigast redan innan man behöver byta lågenergilampan).

Körningsexempel:

Timmar tänd per dag ? 7

Elpris ? 98

Lågenergi billigast efter 149 dagar.

Kommentar: Efter 149 dagar är kostnaden med glödlampor 71.32840 kronor (varav inköpskostnad 10 kronor; man måste byta glödlampa en gång) och med lågenergilampa 71.24354 kronor (varav inköpskostnad 60 kronor).

UPPGIFT 2 – DATUMRÄKNING

Eftersom 2009 inte är ett skottår har det förstås 365 dagar. För att kontrollera detta ville datumräkningsmyndigheten ändå skriva upp alla datum: 1/1, 2/1, 3/1 o.s.v. och räkna hur många olika datum som finns. Tyvärr uppfattade datorprogrammet man använde de inskrivna datumen som bråktal, så exempelvis räknades 1/3, 2/6, 3/9 och 4/12 som samma datum eftersom de motsvarar samma tal. Man kom därför fram till det något sensationella resultatet att året bara har 231 datum, vilket snabbt utlöste en storm av beställningar från kulturer med andra kalendersystem som också ville kontrollera hur många datum man egentligen har. Du ska skriva ett datorprogram som frågar efter antalet månader på året (högst 12) samt antalet dagar i varje månad (högst 40) och beräknar hur många olika datum det finns, förutsatt att man behandlar dem som bråktal.

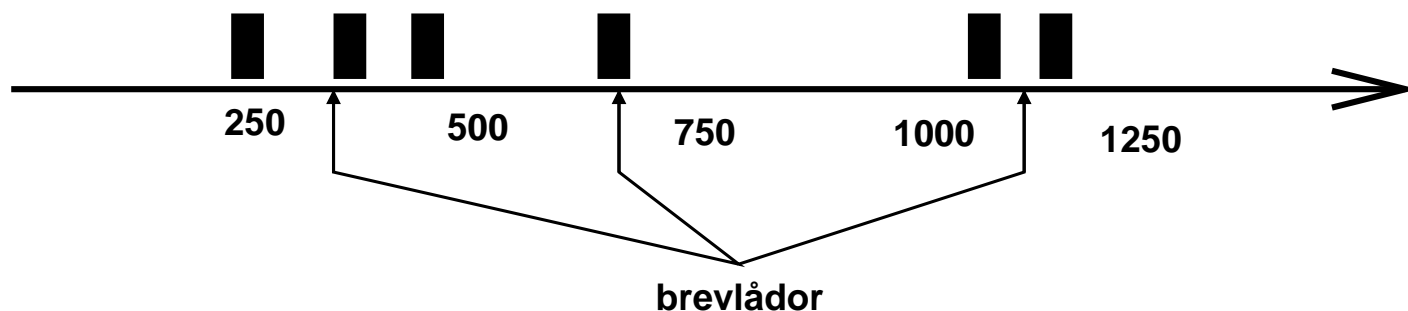
Körningsexempel:

Antal månader ? 6
Dagar i månad 1 ? 2
Dagar i månad 2 ? 5
Dagar i månad 3 ? 3
Dagar i månad 4 ? 6
Dagar i månad 5 ? 5
Dagar i månad 6 ? 2

Antal olika datum: 15

Kommentar: De olika datumen är 1/1, 2/1, 1/2, 3/2, 5/2, 1/3, 2/3, 1/4, 3/4, 5/4, 1/5, 2/5, 3/5, 4/5 och 1/6.

UPPGIFT 3 – BREVLÅDEPLACERING



FIGUR 1. Ett möjligt sätt att placera brevlådorna i exemplet: vid positionerna 360, 690 och 1090.

Utmed en väg ligger hus (högst 20 stycken) utspridda vid givna positioner (angivna i meter mellan 0 och 10000). För att spara tid vid utdelningen vill Posten att husens brevlådor ska grupperas ihop vid speciella platser utmed vägen. Man vill dock garantera att inget hus får längre än 200 meter till sin brevlåda. Du ska skriva ett program som beräknar det minimala antalet brevlådeplatser som behövs.

Körningsexempel:

```

Antal hus ? 6
Position för hus 1 ? 1120
Position för hus 2 ? 256
Position för hus 3 ? 440
Position för hus 4 ? 684
Position för hus 5 ? 380
Position för hus 6 ? 1040
    
```

Antal platser: 3

UPPGIFT 4 – SAMLA MAT

En seriös strategispelare vet hur viktigt det är att samla tillräckligt med mat till en arme innan en invasion ska genomföras. Därför har du bestämt dig för att samla ihop en viss mängd mat till dina trupper.

Från början har du ett visst antal bönder. I varje rond kan varje bonde samla ihop en enhet mat. Efter varje rond kan du sälja en viss mängd mat och köpa en ny bonde. Du kan köpa så många bönder du vill, så länge du har tillräckligt med mat att betala med.

Skriv ett program som beräknar det minsta antalet ronder du måste samla mat för att nå upp till den mängd mat du behöver. Alla värden i indata kommer vara mellan 1 och 1000.

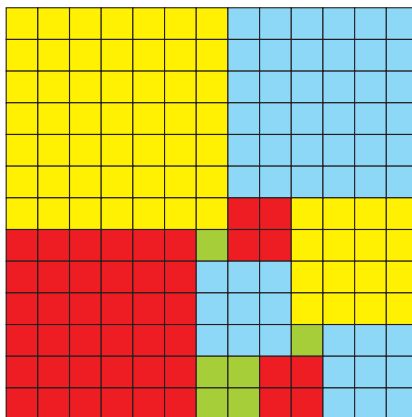
Körningsexempel:

```
Mängd mat du behöver ? 10
Antal bönder från början ? 2
Kostnad för att köpa bonde ? 1
```

Det tar 4 ronder.

Kommentar: Notera att det skulle ta 5 ronder om du inte köpte några bönder. En bättre strategi är att köpa en bonde efter rond 1. Efter rond 1 kommer du då att ha 1 enhet mat, därefter 4 enheter, sedan 7 och slutligen 10.

UPPGIFT 5 – KVADRATPUSSEL



FIGUR 2.

I figur 2 ser du en kvadrat (13×13) som är sammansatt av 11 mindre kvadrater, som också är det minsta antal "småkvadrater" man behöver för att skapa en stor kvadrat av denna storlek.

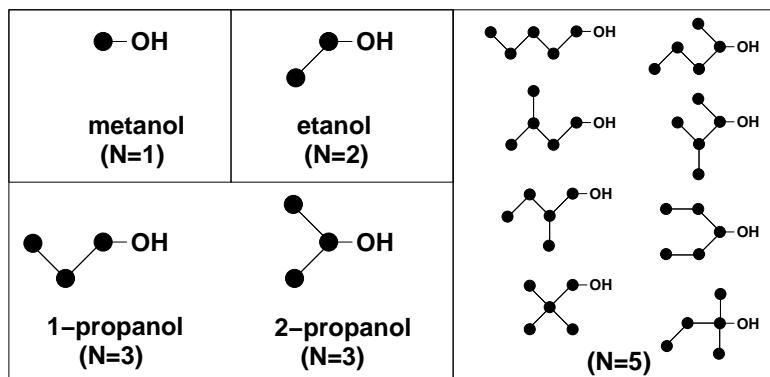
Skriv ett program som tar emot den stora kvadratens sida S , där $2 \leq S \leq 17$, och bestämmer det *minsta* antal mindre kvadrater som behövs för att lägga pusslet.

Körningsexempel:

Kvadratens sida ? 13

Det behövs minst 11 kvadrater

UPPGIFT 6 – ISOMERER



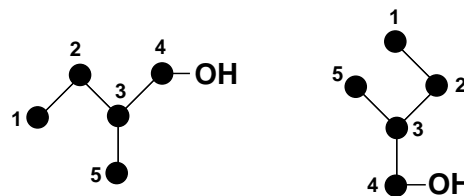
FIGUR 3. Alla isomerer för $N=1$, $N=2$, $N=3$ och $N=5$. För $N=4$ finns fyra isomerer. Varje kula motsvarar en kol-atom och varje streck en bindning.

En molekyl av en viss sorts alkoholer kan beskrivas på följande sätt:

- Den innehåller totalt N kol-atomer.
- Exakt en av dessa är bunden till en s.k. hydroxylgrupp ($-OH$). Denna kol-atom kan binda till högst tre andra kol-atomer.
- Varje övrig kol-atom kan binda till högst fyra andra kol-atomer.
- Alla kol-atomer måste vara sammanlänkade.
- Inga cykler får förekomma.

De enklaste av sådana molekyler är metanol ($N = 1$) och etanol ($N = 2$; "vanlig" alkohol). Från och med $N = 3$ finns det flera olika molekyler som uppfyller kraven ovan men som har olika struktur (se figur ovan). Du ska skriva ett program som frågar efter antalet kolatomer N , där $3 \leq N \leq 11$, och beräknar hur många olika godkända molekyler, s.k. isomerer, det finns.

Två molekyler räknas som olika om det **inte** går att numrera kolatomerna i vardera molekylen på ett sådant sätt att samma par av kol-atomer är sammanbundna i båda molekyllerna och att samma kol-atom är bunden till hydroxylgruppen.



Denna numrering av kol-atomerna avslöjar att detta är två avbildningar av **samma** isomer.

Körningsexempel:

Antal kolatomer ? 6
 Antal isomerer: 17