

Programmeringsolympiaden 2010

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 (IOI) anordnas i Kanada i augusti. Kanske blir du en av dem som representerar Sverige där.

Lycka till!

UPPGIFT 1 – BUSSKORTET



Du ska köpa ett busskort. Det är ett påtankningskort som du laddar med pengar, där-
efter kan du åka på kortet tills pengarna tagit slut. Du vet att du ska åka för K kronor,
där $K \leq 10000$. Att ladda kortet tar sin tid eftersom du endast kan ladda kortet med
500, 200 eller 100 kr i taget. Du har för tillfället bråttom och vill därför utföra så litet
antal transaktioner som möjligt, men aldrig tanka på mer pengar än nödvändigt.

Om du ska åka för 800 kronor ska du alltså först ladda med 500, sen med 200, därefter
med 100 kr. Om du däremot ska åka för 850 kronor ska du först ladda med 500 och
därefter ladda med 200 två gånger. Visserligen går 50 kronor till spillo, men det är ändå
det bästa alternativet.

Skriv ett program som beräknar det minsta antalet transaktioner du behöver göra.

Körningsexempel 1:

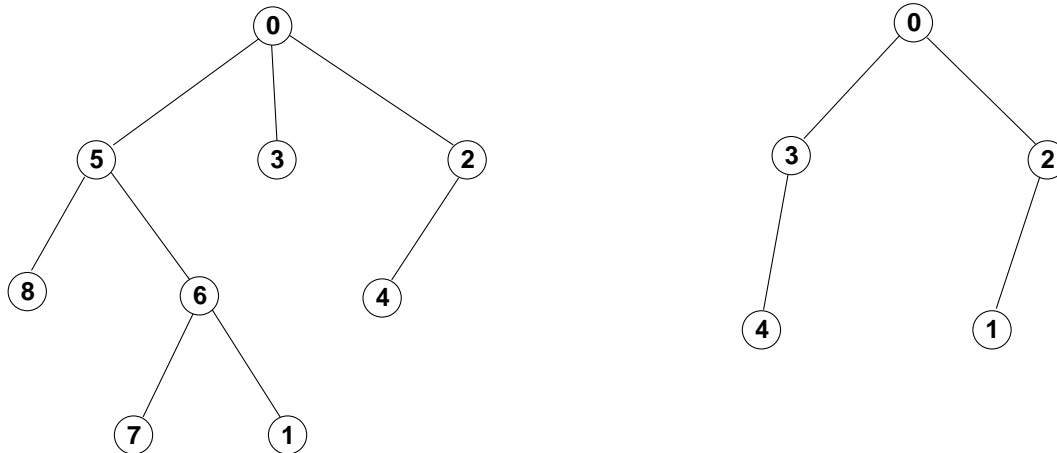
Åka för ? 850
Antal transaktioner: 3

Körningsexempel 2:

Åka för ? 1800
Antal transaktioner: 5

UPPGIFT 2 – SLÄKTTRÄFFEN

Det är släkträff för ättlingar till Ida-Ottilia Isaksson. För enkelhets skull har man upprättat ett släkträd och numrerat alla ättlingarna från 1 till N , samt givit Ida-Ottilia själv numret 0. Bland de M personerna vid ditt bord uppkommer en diskussion om vem som är er närmaste gemensamma släkting. Skriv ett program som räknar ut detta.



FIGUR 1. Släkträden för de två körningsexemplen.

Programmet ska fråga efter antalet ättlingar, N , ($2 \leq N \leq 20$) och därefter fråga efter numret på varje persons förälder, vilket naturligtvis alltid är mellan 0 och N . Därefter ska programmet fråga efter antalet personer vid bordet, M ($2 \leq M \leq N$), och läsa in numret på var och en av dem. Programmet ska skriva ut numret på den person som är närmast gemensam släkting (uppåt i trädet) till alla vid bordet. Observera att detta ibland kan vara någon vid bordet.

Körningsexempel 1:

```

Antal ättlingar ? 8
Förälder till nr 1 ? 6
Förälder till nr 2 ? 0
Förälder till nr 3 ? 0
Förälder till nr 4 ? 2
Förälder till nr 5 ? 0
Förälder till nr 6 ? 5
Förälder till nr 7 ? 6
Förälder till nr 8 ? 5
Antal vid bordet ? 3
Person 1 vid bordet ? 1
Person 2 vid bordet ? 5
Person 3 vid bordet ? 8
    
```

Närmaste släkting: 5

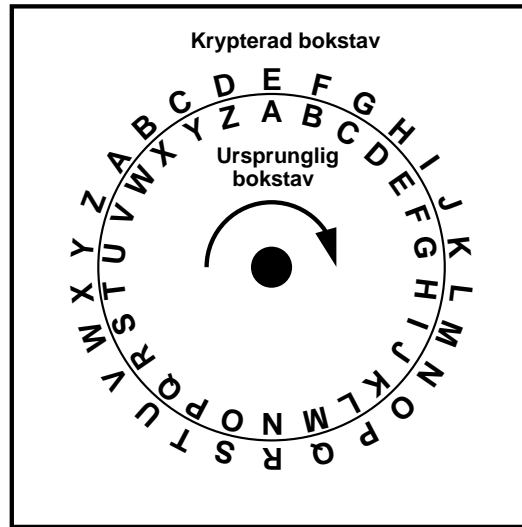
Körningsexempel 2:

```

Antal ättlingar ? 4
Förälder till nr 1 ? 2
Förälder till nr 2 ? 0
Förälder till nr 3 ? 0
Förälder till nr 4 ? 3
Antal vid bordet ? 2
Person 1 vid bordet ? 4
Person 2 vid bordet ? 1
    
```

Närmaste släkting: 0

UPPGIFT 3 – KRYPTERING



FIGUR 2. Bokstavssnurran. De 26 bokstäverna (A-Z) är placerade i ordning (medurs) på båda skivorna. Bilden visar utgångsläget för krypteringen i första exemplet, sedan snurras skivan 9, 12, 15, 18, ... steg för varje kodad bokstav.

En måttligt framgångsrik bataljon använder sig av en variant av Vignère-krypto för att kryptera sina meddelanden. Man använder sig av en bokstavssnurra (se figur ovan) som består av två ringar med bokstäverna i alfabetet (A-Z) där den inre kan förskjutas i förhållande till den yttre. När man krypterar sitt meddelande så letar man rätt på bokstaven på den inre skivan men i den krypterade texten skriver man istället motsvarande bokstav på den yttre skivan.

För att förvilla fienden så snurrar man den inre skivan ett antal steg (medurs) mellan kodningen av varje bokstav. Och för att göra det extra svårt så snurrar man inte lika många steg varje gång utan man ökar antalet steg med ett konstant värde varje gång. Dessutom kan snurran ha vilket läge som helst när man börjar koda.

Du har kommit på ett sätt att knäcka koden: du har nämligen upptäckt att alla meddelanden börjar med HEJ. Skriv ett datorprogram som, givet den krypterade texten skriver ut det ursprungliga meddelandet. Meddelandet innehåller endast versalerna A-Z (alltså t.ex. inga blanksteg) och är mellan 3 och 20 bokstäver långt.

Körningsexempel 1:

Krypterad text ? LRIJOUZRIYAQIRAG
 Meddelande: HEJVITARENFIKANU

Körningsexempel 2:

Krypterad text ? GCGDJJI
 Meddelande: HEJHOPP

UPPGIFT 4 – FERMATS LEK

$$A^3 = B^3 + C^3 + D^3 + \dots$$

Skriv ett program, som givet ett heltal A , där $2 \leq A \leq 100$, hittar det minsta antalet heltal B, C, D, \dots , där $1 \leq B, C, D, \dots < A$, så att ekvationen ovan uppfylls. Enligt Fermats stora sats¹ finns det aldrig någon lösning med enbart två termer. Om det finns flera lika korta lösningar ska programmet skriva ut vilken som helst av dem och ordningen på talen spelar ingen roll.

Körningsexempel 1:

A ? 6
Tal: 3 4 5

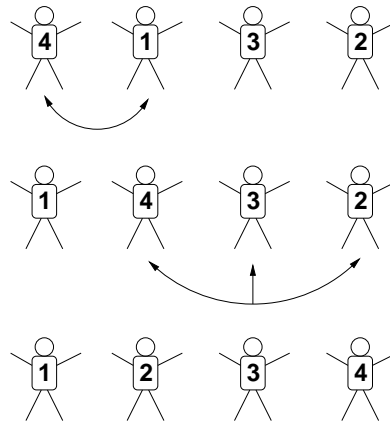
Körningsexempel 2:

A ? 8
Tal: 2 2 4 6 6

¹Satsen gäller alla exponenter större än 2 och bevisades av Andrew Wiles så sent som 1995. Specialfallet med exponent 3 bevisades dock av Euler redan 1770.

UPPGIFT 5 – UPPSTÄLLNING

N personer, numrerade från 1 till N , står uppställda huller om buller på en rad. Du ska få personerna att stå i nummerordning (1, 2, 3, ..., N) genom att ge ett antal kommandon. Med varje kommando omvänder du ordningen på ett valfritt antal intill varandra stående personer. Skriv ett program som beräknar det minsta antalet kommandon du behöver ge.



FIGUR 3. Den översta raden visar utgångsställningen i det första körningsexemplet, den mellersta raden ställningen efter ett kommando och den nedersta raden slutställningen.

Programmet ska fråga efter antalet personer N , där $2 \leq N \leq 8$, och sedan efter utgångsställningen. Du kan förutsätta att de inmatade talen är mellan 1 och N och att varje tal förekommer exakt en gång. Programmet ska skriva ut det minsta antalet kommandon som krävs för att ställa personerna i nummerordning.

Körningsexempel 1:

Antal personer ? 4
 Person på position 1 ? 4
 Person på position 2 ? 1
 Person på position 3 ? 3
 Person på position 4 ? 2

Minsta antal kommandon: 2

Körningsexempel 2:

Antal personer ? 2
 Person på position 1 ? 1
 Person på position 2 ? 2

Minsta antal kommandon: 0

Körningsexempel 3:

Antal personer ? 5
 Person på position 1 ? 3
 Person på position 2 ? 5
 Person på position 3 ? 1
 Person på position 4 ? 4
 Person på position 5 ? 2

Minsta antal kommandon: 3

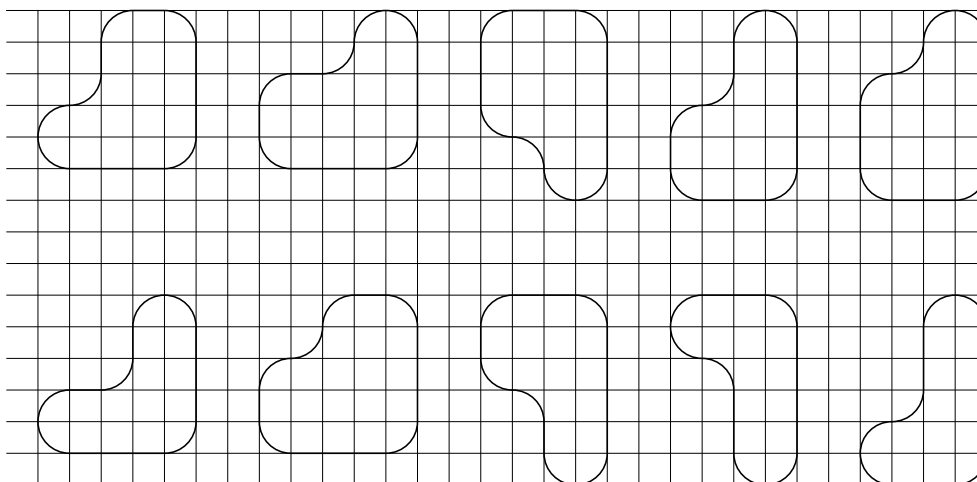
UPPGIFT 6 – TRÄJÄRNVÄGEN

Träjärnväg är en klassisk leksak. I denna uppgift används en enkel variant där det finns två sorters bitar:

- Raka bitar med längden R
- Kurvbitar som motsvarar exakt en fjärdedel av en cirkel med radien R . Dessa är vändbara så att de kan svänga såväl åt höger som åt vänster.

Nora är tre år och har tröttnat på att dra sitt tåg fram och tillbaka. Det hon funderar över är hur många olika sätt man kan bygga ihop en given uppsättning bitar till en tåg bana. Hjälp henne genom att skriva ett datorprogram som räknar ut svaret.

En tåg bana måste innehålla alla bitarna och den måste sitta ihop så att tåget kan köra runt. Den får inte korsas sig själv eller ens ha någon punkt eller sträcka gemensam. Två tågbanor räknas som identiska om man kan starta ett tåg på vardera banan (valfri plats och riktning) och tågen då "upplever" precis samma serie av kurvor och raksträckor då de kör ett varv. Sådana tågbanor ska alltså inte räknas dubbelt. Det totala antalet bitar överstiger aldrig 18.

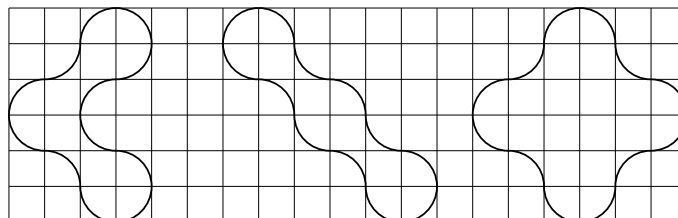


Körningsexempel 1 (figuren ovan):

Antal raka bitar? 8
 Antal kurvbitar? 6
 Antal tågbanor: 10

Körningsexempel 2 (figuren nedan):

Antal raka bitar? 0
 Antal kurvbitar? 12
 Antal tågbanor: 3



Fler exempel på nästa sida...

Körningsexempel 3:

Antal raka bitar? 2
Antal kurvbitar? 6
Antal tågbanor: 0

Körningsexempel 4:

Antal raka bitar? 4
Antal kurvbitar? 6
Antal tågbanor: 1

Körningsexempel 5:

Antal raka bitar? 2
Antal kurvbitar? 8
Antal tågbanor: 6