

# Programmeringsolympiaden 2013

## TÄVLINGSREGLER FÖR SKOLKVALET

- Tävlingen äger rum på av skolan bestämt datum under sex timmar effektiv tid.
- Tävlingen består av sju uppgifter som alla 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.
- Första uppgiften är gemensam för skolkvalet och onlinekvalet och kan därför alternativt lösas via vår rättningsdomare Kattis, senast söndag 3 februari. För mer information, se [www.progolymp.se](http://www.progolymp.se)
- Samtliga uppgifter leder fram till program vars exekveringstid normalt bör understiga 1 sekund. Om programmets 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: Valfritt skriftligt material samt de manualfiler som är installerade på datorn. Däremot är det inte tillåtet att söka information på eller kommunicera via internet. Räknedosa är tillåten.
- Tävlingsbidraget ska lämnas in i form av källkodsfiler som läggs på utdelat minne eller i en av läraren angiven hårddiskkatalog. Filerna ska döpas till uppg1...uppg7 med passande filtillägg. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

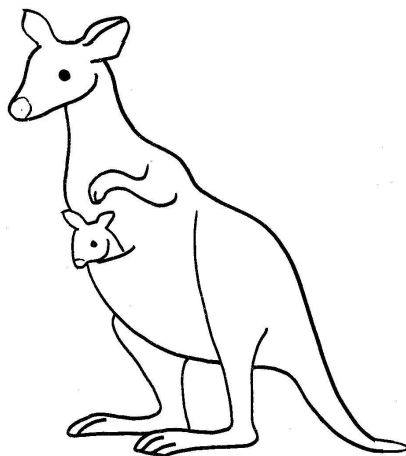
Årets International Olympiad in Informatics (IOI) anordnas i Brisbane, Australien, i juli. Kanske blir du en av dem som representerar Sverige där.

**Lycka till!**

## UPPGIFT 1 – KÄNGURUMAMMAN

*OBS! Denna uppgift är "öppen" och kan även lösas i efterhand via vår hemsida:*

**[www.progolymp.se](http://www.progolymp.se)**



En kängurumamma ska packa ner sina barn i sin pung. Hon har massor med barn. De två minsta väger bara ett gram var men sen väger varje barn lika mycket som de två föregående tillsammans. Hennes sex minsta barn väger alltså 1, 1, 2, 3, 5 och 8 gram.

Mamman orkar bära högst  $X$  kilo (och inte ens ett gram mer). Hur många barn kan hon som mest ta med sig? I samtliga tester kommer  $X$  vara ett heltal mellan 1 och 1000.

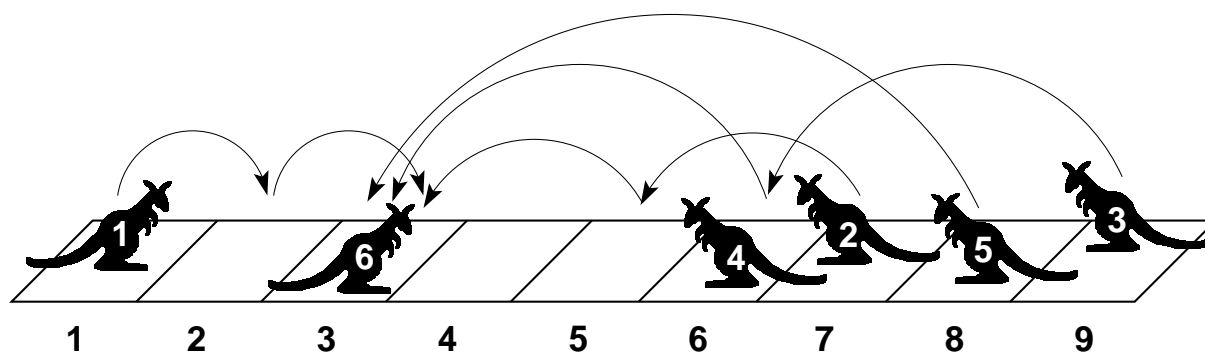
Delpoäng ges för svaret på specialfallet  $X = 1$ , även om program inte inlämnas.

### **Körningsexempel**

Talet  $X$  ? 4

Antal barn: 16

## UPPGIFT 2 – HOPPLEKEN



FIGUR 1. Startställningen i första exemplet. Pilarna visar hur alla barnen utom nummer 4 kan samlas på ruta 3.

När kängurumammans barn blivit äldre och står på egna ben, leker de en hopplek (för vad leker kängurur annars?). De har ritat upp  $N$  rutor (högst 100) i en rad på marken, numrerade från 1 till  $N$ . Eftersom barnen (högst 10 stycken) är olika stora hoppar de olika långt: det första (minsta) barnet hoppar alltid en ruta i taget, det andra barnet alltid två rutor i taget, det tredje barnet alltid tre rutor i taget o.s.v.

Givet i vilken ruta varje kängurubarn står från början (alla olika), skriv ett program som räknar ut det maximala antalet barn som kan hamna i samma ruta. De kan göra hur många hopp de vill och de kan även stå stilla, men de får inte hoppa utanför rutorna.

### Körningsexempel 1

```

Antal barn ? 6
Antal rutor ? 9
Position för barn 1 ? 1
Position för barn 2 ? 7
Position för barn 3 ? 9
Position för barn 4 ? 6
Position för barn 5 ? 8
Position för barn 6 ? 3
Maximal samling: 5
    
```

### Körningsexempel 2

```

Antal barn ? 4
Antal rutor ? 5
Position för barn 1 ? 4
Position för barn 2 ? 2
Position för barn 3 ? 3
Position för barn 4 ? 5
Maximal samling: 2
    
```

## UPPGIFT 3 – LEET SPEAK

Pappa Thomas har tröttnat. Sonen Pim gör inget annat än att sitta framför datorn och spela skjutspel. Han spelar så mycket skjutspel att han inte bryr sig om att prata med sin pappa längre. Pappa Thomas har bestämt sig för att försöka ta kontakt med Pim genom att logga in på det där internet och ta reda på vad sonen egentligen sysslar med. För att det inte ska verka misstänkt så vill han prata samma språk som Pims cyberkompisar gör. Han har förstått att språket kallas leet speak, men han behärskar det inte själv. Thomas tog först kontakt med ett konsultbolag och bad dem skriva ett program som översatte meningar från vanligt språk till leet speak. Konsultbolaget ansåg dock att problemet var för svårt.

Därför tar nu Thomas det här till nästa nivå, och vänder sig till de tävlade i Programmeringsolympiaden. Kan du hjälpa honom att skriva ett program som översätter meningar från vanligt språk till leet speak? Leet speak är ett substitutionschiffer där varje bokstav i det vanliga alfabetet byts ut mot ett tecken i leet speak-alfabetet.

Till din hjälp har du alfabetet för respektive språk:

```
vanligt:      abcdefghijklmnopqrstuvwxyz
leet speak:  48()3}6#!]X|MN09Q2Z7MVWXJZ
```

Programmet ska läsa in en *mening*: en rad med tecken som kan vara a-z (stora eller små bokstäver), blanksteg eller punkt. Raden avslutas alltid med en punkt, och det finns inga andra punkter i raden. Blanksteg förekommer aldrig i början av meningen och aldrig mer än ett i följd. Programmet ska skriva ut meningen översatt till leet speak.

### Körningsexempel 1

Vanligt ? lEet speAK.

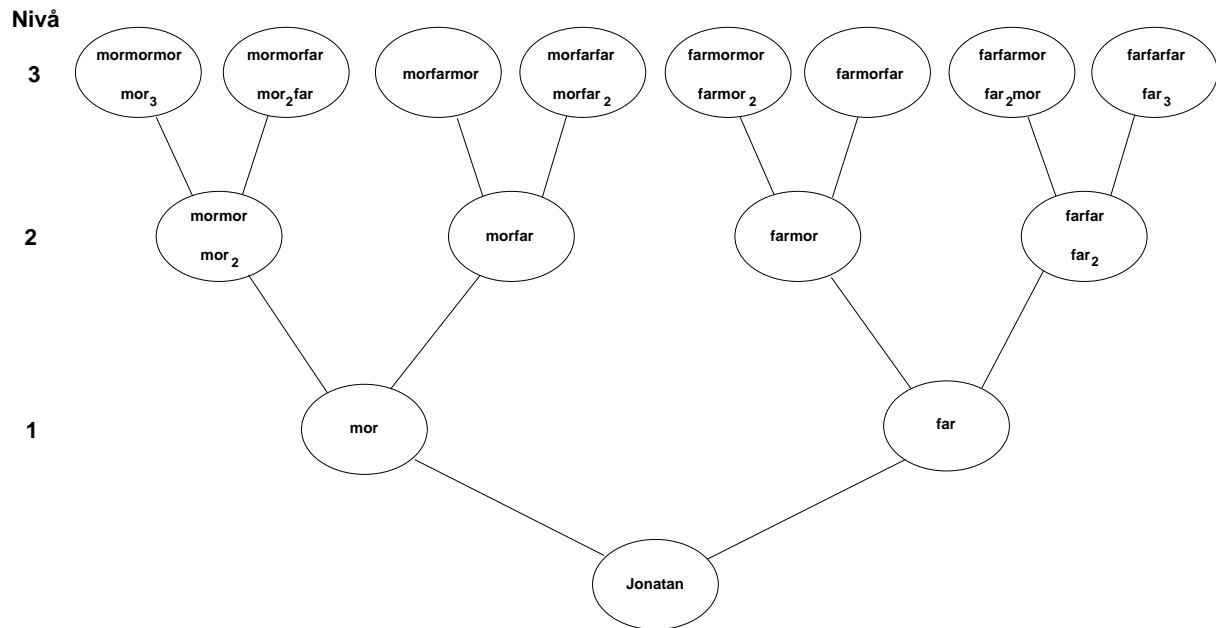
Översatt: |337 Z934X.

### Körningsexempel 2

Vanligt ? The quick brown fox jumps over a lazy dog.

Översatt: 7#3 QM!(X 820WN }0X ]MM9Z 0V32 4 |4ZJ )06.

## UPPGIFT 4 – MOR OCH FAR



Jonatan har insett att man kan referera hur långt bak som helst i släktträdet genom att kombinera “mor” och “far”. Sina föräldrar, som vi kallar nivå 1, kan man skriva *mor* och *far*, därefter på nivå 2 kan man skriva *mormor*, *morfar*, *farmor* och *farfar* och så vidare. Efter ett tag blir orden ganska långa, men de kan förkortas om man tillåter lite matematik att komma in i språket. Vi skriver exempelvis om *mormormorfarfarmorfar* till *mor<sub>3</sub>far<sub>2</sub>morfar*.

Jonatan vill skriva en lista över alla sina förfäder, upp till och med en viss nivå (mellan 1 och 18), på denna form. Hur många heltal behöver han skriva?

### Körningsexempel 1

Nivå ? 3

Antal: 8

### Körningsexempel 2

Nivå ? 1

Antal: 0

### Körningsexempel 3

Nivå ? 10

Antal: 4608

## UPPGIFT 5 – EKVATIONER

Ekvationer är viktiga inom matematiken och redan på högstadiet får man lära sig att lösa enklare typer. I den här uppgiften så ska du göra detsamma, men nu med hjälp av datorn.

Du ska skriva ett program som läser in en ekvation och löser den. Ekvationerna består av högst 20 tecken och innehåller för enkelhets skull endast ensiffriga tal (0–9), tecknen +, −, = och tecknet  $x$ , som är den okända variabeln.

Det garanteras att varje ekvation innehåller precis ett likhetstecken, och att ekvationerna i övrigt är syntaktiskt korrekta. Närmare bestämt består de av ett antal *termer* (*siffra* eller  $x$  eller *siffra*  $x$ ) och termerna åtskiljs av exakt ett av tecknen +, − eller =.

Programmet ska skriva ut det värde på  $x$  som löser ekvationen (minst sex decimaler vid behov). Om det inte finns något sådant värde, skriv istället ut “Ingen lösning”. Om det tvärtom finns oändligt många värden som löser ekvationen, skriv ut “Flera lösningar”.

Delpoäng kommer ges om programmet klarar alla fall med en unik lösning, eller om programmet klarar alla övriga fall.

### Körningsexempel 1

Ekvation ?  $3x+2=8-x$   
 $x = 1.5$

### Körningsexempel 2

Ekvation ?  $0=8+3x$   
 $x = -2.666667$

### Körningsexempel 3

Ekvation ?  $x+7-9x=2+5x-3$   
 $x = 0.615385$

### Körningsexempel 4

Ekvation ?  $3x+5=5-4x$   
 $x = 0$

### Körningsexempel 5

Ekvation ?  $6x=6x$   
Flera lösningar

### Körningsexempel 6

Ekvation ?  $x+1=x-1$   
Ingen lösning

## UPPGIFT 6 – SOKOBAN

I spelet Sokoban rör du dig i ett lagerrum indelat i rutor och ska flytta ett antal lådor (som upptar exakt en ruta) från sina startpositioner (S) till sina målpositioner (M). Följande regler gäller:

- Du (D) kan bara röra sig vänster, höger, upp eller ned i rutnätet, alltså inte diagonalt.
- Du kan flytta en låda endast genom att skjuta den framför dig.
- Endast en låda i taget kan flyttas.
- Du kan inte gå genom lådor eller väggar (#) och inte heller skjuta en låda genom vägg eller annan låda.
- Spelet är löst när alla lådor hamnat på målpositioner. Det spelar ingen roll vilken låda som hamnar på vilken målposition.

Du ska skriva ett program som spelar Sokoban. Programmet ska fråga efter antal rader (högst 5) och antal kolumner (högst 5) och sedan läsa in en karta över lagerrummet där '.' betyder tom ruta, '#' betyder blockerad ruta (vägg), 'S' betyder startposition för låda, 'M' betyder målposition för låda och 'D' betyder din startposition. Det finns alltid lika många S som M (högst 5 stycken) och det finns alltid exakt ett D. Programmet ska skriva ut en rad med tecken (högst 1000) som beskriver hur du ska röra dig för att uppnå målet (H=höger, V=vänster, U=upp, N=ner). Programmet behöver inte hitta den kortaste dragföljden. I alla testfall kommer det att finnas en dragföljd med högst 20 drag.

### Körningsexempel 1

Antal rader: 1  
Antal kolumner: 4  
Rad 1 ? #MSD

Dragföljd: V

### Körningsexempel 2

Antal rader: 4  
Antal kolumner: 3  
Rad 1 ? #..  
Rad 2 ? .SD  
Rad 3 ? .#M  
Rad 4 ? ...

Dragföljd: NNVVUUHUHN

### Körningsexempel 3

Antal rader: 3  
Antal kolumner: 4  
Rad 1 ? .S.M  
Rad 2 ? ..SD  
Rad 3 ? MMS.

Dragföljd: NVVUVUHHNHNV

**Kommentar:** Samtliga dessa dragföljder är de kortast möjliga.

## UPPGIFT 7 – TÄRNINGSSUMMA

Tärningen är en klassisk symbol för slump. Slumpen är i teorin rättvis, men det gäller tyvärr inte i praktiken. I synnerhet gäller det inte i denna uppgift, där vi jobbar med *orättvisa tärningar*.

Du kastar ett antal ( $1 \leq N \leq 100$ ) identiska sexsidiga tärningar exakt en gång var. Du känner till sannolikheten  $p(i)$ ,  $i = 1, 2, 3, 4, 5, 6$  för att med en tärning få etta, tvåa, trea o.s.v. (du kan anta att sannolikheterna adderar upp till exakt ett).

Skriv ett program som beräknar sannolikheten för att summan av tärningarnas utfall blir större än eller lika med ett visst heltalsvärde  $S$ , där  $1 \leq S \leq 600$ . Svaret ska vara korrekt med minst 6 decimalers noggrannhet.

Delpoäng kommer ges om programmet klarar alla testfall med  $N \leq 10$ .

### Körningsexempel 1

```
Antal tärningar ? 2
Talet S ? 6
p(1) ? 0.2
p(2) ? 0.2
p(3) ? 0.2
p(4) ? 0.2
p(5) ? 0.1
p(6) ? 0.1
```

Svar: 0.6

### Körningsexempel 2

```
Antal tärningar ? 100
Talet S ? 353
p(1) ? 0.1
p(2) ? 0.2
p(3) ? 0.15
p(4) ? 0.3
p(5) ? 0.15
p(6) ? 0.1
```

Svar: 0.432678