

# Programmeringsolympiaden 2012

## Kvalificering

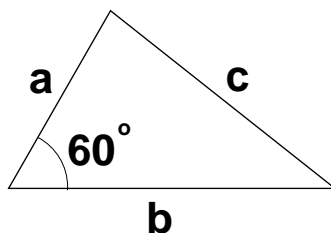
### TÄVLINGSREGLER

- 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: 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 i roten på utdelat minne eller i en av läraren angiven hårddiskkatalog. 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 Italien i september. Kanske blir du en av dem som representerar Sverige där.

## Lycka till!

## UPPGIFT 1 – LAGOMVINKLADE TRIANGLAR



FIGUR 1. Ett exempel på en lagomvinklad triangel med sidlängderna  $a = 5$ ,  $b = 8$  och  $c = 7$ .

En *lagomvinklad* triangel är vad vi i denna uppgift kallar en triangel där minst en av vinklarna är exakt 60 grader. De lagomvinklade trianglarna känner sig ofta förbisedda jämfört med de mycket mer kända rätvinkligna trianglarna (så kallat mindervinkelkomplex), trots att de lagomvinklade också har en snygg formel för sina sidlängder:

$$c^2 = a^2 + b^2 - ab$$

Skriv ett program som skipar lite rättvisa i detta triangeldrama genom att fråga efter ett tal  $N$  (mellan 1 och 100) och sedan skriva ut hur många lagomvinklade trianglar det finns vars sidor är heltal i intervallet 1 till  $N$ .

### Körningsexempel 1

Talet  $N$  ? 25

Antal trianglar: 35

*Förklaring: Först finns det 25 liksidiga trianglar med sidlängder 1 till 25. De övriga är:*

3 8 7  
5 8 7  
5 21 19  
6 16 14  
7 15 13  
8 15 13  
9 24 21  
10 16 14  
15 24 21  
16 21 19

### Körningsexempel 2

Talet  $N$  ? 70

Antal trianglar: 112

## UPPGIFT 2 – WORDFEUD

	1	2	3	4	5	6	7	8	9	10
1										
2							DL			
3						TW				
4										
5				TL		DL				
6							DW			
7										
8										
9										
10										

FIGUR 2. Spelplanen i exemplet med specialrutorna utmärkta. De skuggade områdena visar tre exempel på placering av "ordet" 56141. Den lodräta placeringen ger  $2*(2*5+6+1+2*4+1)=52$  poäng, den vågräta på rad 3 ger 51 poäng och den vågräta på rad 5 ger 33 poäng.

Wordfeud är ett scrabble/alfabet-liknande spel där man ska lägga ut bokstavsbrickor på en rutig spelplan så att det bildas ord. Orden kan läggas antingen från vänster till höger eller uppifrån och ned. Varje bokstav ger en viss poäng. Dessutom finns ett antal specialrutor på spelplanen, som kan vara av fyra typer:

- DL (double letter): bokstavspoängen för en bricka på denna ruta dubblas
- TL (triple letter): bokstavspoängen för en bricka på denna ruta tredubblas
- DW (double word): poängen för hela ordet dubblas om rutan täcks
- TW (triple word): poängen för hela ordet tredubblas om rutan täcks

När man räknar ut poängen för ett ord börjar man med att summera bokstavspoängen, med eventuell multiplikation på varje bokstav. När summan är uträknad multipliceras den med eventuella ordmultiplikatorer. Observera att om ordet ligger på flera DW- eller TW-rutor kan poängen multipliceras flera gånger.

I den här uppgiften tänker vi oss en tom 10x10 spelplan där raderna och kolumnerna är numrerade enligt figuren. Skriv ett program som, givet specialrutornas placering på spelplanen och det ord du tänker lägga ut, beräknar den högsta poäng du kan få om du får lägga ordet var du vill (men hela ordet inom spelplanen).

Programmet ska fråga efter antalet specialrutor (högst 10 stycken) och sedan läsa in deras koordinater och typ enligt dialogen nedan. Både raden och kolumnen ligger i intervallet 1 till 10 och typen är alltid DL, DW, TL eller TW. Slutligen ska programmet fråga efter ordet som ska läggas. För enkelhets skull har varje bokstav redan ersatts med sitt poängvärde, en siffra mellan 0 och 8. Ordet kan ha upp till 7 bokstäver.

### Körningsexempel 1

Antal specialrutor ? 5  
Specialruta 1, rad ? 2  
Specialruta 1, kolumn ? 7  
Specialruta 1, typ ? DL  
Specialruta 2, rad ? 5  
Specialruta 2, kolumn ? 5  
Specialruta 2, typ ? TL  
Specialruta 3, rad ? 6  
Specialruta 3, kolumn ? 7  
Specialruta 3, typ ? DW  
Specialruta 4, rad ? 5  
Specialruta 4, kolumn ? 7  
Specialruta 4, typ ? DL  
Specialruta 5, rad ? 3  
Specialruta 5, kolumn ? 6  
Specialruta 5, typ ? TW  
Ord ? 56141

Maxpoäng: 52

I följande exempel har dialogen utelämnats för att spara plats på sidan.

### Körningsexempel 2

4  
1 9 DW  
1 10 DL  
2 9 TW  
2 10 TL  
174

Maxpoäng: 72

### Körningsexempel 3

9  
1 1 DL  
1 4 DW  
3 1 TW  
3 4 TL  
2 3 TL  
2 8 TW  
9 3 DW  
9 8 DW  
9 2 DL  
2152652

Maxpoäng: 100

### UPPGIFT 3 – TALFAMILJER

Följande uppgift är en generalisering av ett problem från det senaste kvalet till skolornas matematiktävling. Vi säger att varje positivt heltal  $N$  har en familj som består av  $N$  samt alla positiva heltal man kan få genom att ordna om  $N$ 's siffror, utom dem som vid omordningen får en nolla som första siffra. (T.ex. har talet 101 familjen 101, 110.) Vi säger också att  $N$ 's familj gillar det positiva heltalet  $p$  om  $N$  eller något annat tal i familjen är delbart med  $p$ . (Alla tal som familjen ovan gillar är 1,2,5,10,11,22,55,101,110.)

Skriv ett program som, givet ett antal (högst 10) positiva heltal, bestämmer det minsta positiva heltalet vars familj gillar samtliga av de givna talen. I givna testfall kommer det alltid att finnas ett sådant tal med högst sex siffror.

**Körningsexempel 1** (detta var uppgiften i mattetävlingen)

```
Antal tal? 5
Tal 1 ? 3
Tal 2 ? 5
Tal 3 ? 7
Tal 4 ? 9
Tal 5 ? 11
```

Svar: 459

**Körningsexempel 2**

```
Antal tal? 3
Tal 1 ? 79
Tal 2 ? 97
Tal 3 ? 113
```

Svar: 1469

**Körningsexempel 3**

```
Antal tal? 7
Tal 1 ? 164
Tal 2 ? 278
Tal 3 ? 293
Tal 4 ? 382
Tal 5 ? 483
Tal 6 ? 598
Tal 7 ? 23
```

Svar: 102246

## UPPGIFT 4 – BOKHYLLOR

Du ska köpa bokhyllor för att få plats med alla dina böcker. Du vet från början vilka böcker du har och behöver räkna ut antalet bokhyllor som krävs. Böckerna har tre olika storlekar: en liten bok tar 1 platsenhet, en mellanstor 2 och en stor bok tar 3 platsenheter. Varje hylla rymmer ett visst antal platsenheter. Givet hur många böcker du har av varje sort, skriv ett program som beräknar hur många hyllor du behöver om du vill ha så få hyllor som möjligt. Antalet böcker av varje sort kommer vara högst 20.

### Körningsexempel 1

Antal små ? 30  
Antal mellan ? 0  
Antal stora ? 0  
Hyllstorlek ? 10

Antal hyllor: 3

*Du sätter 10 små böcker på varje hylla.*

### Körningsexempel 2

Antal små ? 0  
Antal mellan ? 0  
Antal stora ? 10  
Hyllstorlek ? 10

Antal hyllor: 4

*Du sätter 3 stora böcker på 3 hyllor, och får tyvärr en hylla till med bara en bok i.*

### Körningsexempel 3

Antal små ? 8  
Antal mellan ? 7  
Antal stora ? 5  
Hyllstorlek ? 15

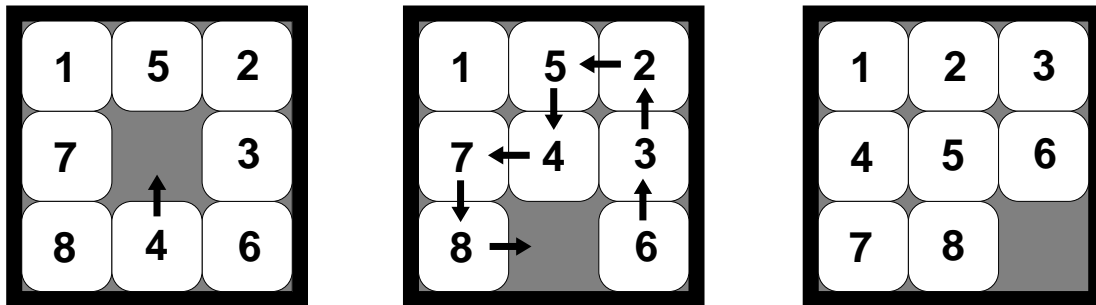
Antal hyllor: 3

### Körningsexempel 4

Antal små ? 1  
Antal mellan ? 4  
Antal stora ? 13  
Hyllstorlek ? 7

Antal hyllor: 8

UPPGIFT 5 – ÅTTASPELET



FIGUR 3. Till vänster startpositionen i det första exemplet. Pilen visar det första draget. Bilden i mitten visar ställningen efter det första draget och pilarna indikerar de återstående sju dragen som måste göras för att nå slutställningen som visas till höger.

*Åttaspelet* (ett småsyskon till det mer kända *femtonspelet*) är en typ av pussel som består av åtta kvadratiska brickor i en ram. Brickorna är numrerade från 1 till 8. Ramen rymmer  $3 \times 3 = 9$  brickor, så det finns ett hål. Ett drag består i att man skjuter en intilliggande bricka till hålets plats. Skriv ett program som givet en startställning beräknar det minsta antalet drag som behövs för att uppnå den ordnade slutställningen som visas till höger i figuren ovan.

Programmet ska läsa in bricknumret på var och en av de 9 positionerna, eller 0 för hålet. Numren matas in i vanlig "läsordning" (översta raden först, från vänster till höger). I givna testfall kommer det alltid vara möjligt att nå slutställningen.

Tips: En godtycklig ställning är antingen olöslig (vilket inte förekommer i testfallen) eller har en lösning med högst 31 drag.

**Körningsexempel 1**

Bricka ? 1  
 Bricka ? 5  
 Bricka ? 2  
 Bricka ? 7  
 Bricka ? 0  
 Bricka ? 3  
 Bricka ? 8  
 Bricka ? 4  
 Bricka ? 6

Antal drag: 8

**Körningsexempel 2**

Bricka ? 3  
 Bricka ? 7  
 Bricka ? 8  
 Bricka ? 5  
 Bricka ? 4  
 Bricka ? 6  
 Bricka ? 2  
 Bricka ? 0  
 Bricka ? 1

Antal drag: 27

## UPPGIFT 6 – FOTBOLLSMATCHEN

Två lag ska spela en fotbollsmatch, och vi ska räkna ut sannolikheten att lag 1 vinner över lag 2.

Fotbollsplanen som de spelar på är inte symmetrisk, utan målen är olika stora, så vilken planhalva man har kan vara avgörande för resultatet. Samtidigt är givetvis inte nödvändigtvis heller de två lagen exakt lika starka. Lagen spelar tills något av lagen har gjort  $n$  mål och då har detta lag vunnit. Du får givet i indata sannolikheten att lag 1 gör nästa mål. Denna sannolikhet beror endast på vilken planhalva laget spelar på.

Spelarna kom på att man kan kompensera orättvisan med planhalvorna genom att växelvis byta planhalva. Man bestämde att efter att något lag gjort 5 mål byter man planhalva, sedan byter man tillbaka vid 10, återigen vid 15, 20, 25 o.s.v.. Lag 1 börjar alltid spela på planhalva 1. Notera att man alltså granskar maxvärdet av lagens mål, inte summan! T.ex. byter man inte då det blir (3-2), utan vid t.ex. (5-2). Inte heller vid (5-5), utan först vid t.ex. (7-10).

Programmet ska fråga efter heltalet  $n$  (där  $1 \leq n \leq 100$ ), antalet mål ett lag måste göra innan det vinner. Därefter ska det fråga efter flyttalet  $P_1$ , sannolikheten att lag 1 gör nästa mål när de har planhalva 1, och slutligen efter flyttalet  $P_2$ , sannolikheten att lag 1 gör nästa mål när de har planhalva 2. Självklart gäller att  $0 \leq P_1, P_2 \leq 1$ . Observera att sannolikheten att lag 2 gör nästa mål lätt kan beräknas: den är  $1 - P_1$  eller  $1 - P_2$ , beroende på om lag 1 spelar på planhalva 1 eller 2. Programmet ska skriva ut sannolikheten att lag 1 vinner matchen. Formatet spelar ingen roll men svaret ska vara korrekt med minst 6 decimaler.

### Körningsexempel 1

$n$  ? 1  
 $P_1$  ? 0.55  
 $P_2$  ? 1.0

Svar: 0.55

*Här spelar det ingen roll att lag 1 är överlägsna på den andra planhalvan, i och med att de bara spelar tills det blivit ett mål.*

### Körningsexempel 2

$n$  ? 6  
 $P_1$  ? 0.55  
 $P_2$  ? 1.0

Svar: 1.0

*Även om lag 2 gör 5 raka mål kommer de hur som helst byta planhalva sedan, och då kan endast lag 1 göra mål och garanterat vinna.*

### Körningsexempel 3

$n$  ? 17  
 $P_1$  ? 0.75  
 $P_2$  ? 0.3

Svar: 0.5819179