

Programmeringsolympiaden 2011

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.
- Hjälpmiddel: 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 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 Thailand i juli. Kanske blir du en av dem som representerar Sverige där.

Lycka till!

UPPGIFT 1 – RECEPTET

Du har bestämt dig för att laga mat. För att laga maten behöver du N ingredienser. För varje ingrediens vet du hur mycket du redan har hemma, hur mycket du behöver totalt samt kostnaden för ingrediensen om du måste köpa den. Du skall alltså köpa den mängd av varje ingrediens som du saknar. Uppgiften är att beräkna kostnaden för att laga maten. Det kommer inte finnas mer än 5 ingredienser.

Programmet ska fråga efter antalet ingredienser och sedan för varje ingrediens hur mycket du redan har hemma, hur mycket du behöver totalt samt hur mycket den kostar (per enhet) om du måste köpa den. Programmet ska skriva ut den totala kostnaden.

Körningsexempel:

```
Antal ingredienser ? 3
Nr 1: Hur mycket har du ? 10
Nr 1: Hur mycket behövs ? 5
Nr 1: Vad kostar den ? 1000
Nr 2: Hur mycket har du ? 0
Nr 2: Hur mycket behövs ? 4
Nr 2: Vad kostar den ? 5
Nr 3: Hur mycket har du ? 2
Nr 3: Hur mycket behövs ? 5
Nr 3: Vad kostar den ? 1
```

Total kostnad: 23

Förklaring: Den första ingrediensen som är väldigt dyr har du lyckligtvis redan tillräckligt av. Den andra ingrediensen får du köpa 4 stycken av för 5 kr styck. Den sista ingrediensen köper du endast 3 av, för du har redan 2 och behöver 5, de kostar endast 1 kr styck. Totalt blir notan 23 kr.

UPPGIFT 2 – SIFFERLEK

Termen *digital root* för heltalet N innebär att man upprepade gånger summerar talets ingående siffror tills summan blir < 10 . Till exempel för talet 34783 får man

$$3 + 4 + 7 + 8 + 3 = 25$$

$$2 + 5 = 7$$

Talet 34783 har digital root 7.

På liknande sätt definieras *multiplicative digital root* för heltalet N , där man istället upprepade gånger multiplicerar ingående siffror tills produkten blir < 10 . Till exempel för talet 34783 får man

$$3 \cdot 4 \cdot 7 \cdot 8 \cdot 3 = 2016$$

$$2 \cdot 0 \cdot 1 \cdot 6 = 0$$

Talet 34783 har multiplicative digital root 0. Hos talet 34783 är alltså multiplicative digital root och digital root olika.

Skriv ett program som tar reda på hur många tal i ett givet intervall som har samma digital root som multiplicative digital root. Intervallet ryms alltid mellan 1 och en miljon.

Körningsexempel 1:

Från talet ? 1000

Till talet ? 2000

Antal samma: 33

(Bland dessa finns till exempel 1124, 1355, 1473 och 1977.)

Körningsexempel 2:

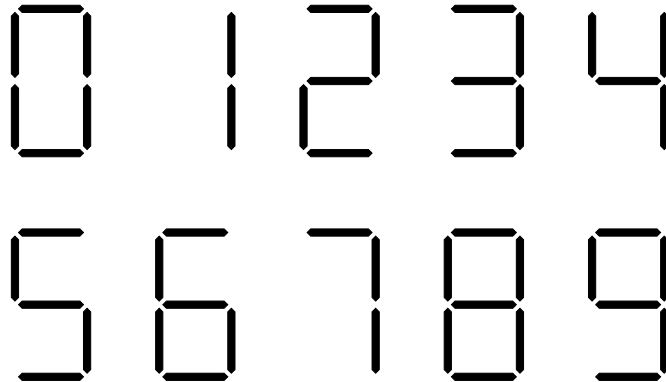
Från talet ? 364

Till talet ? 371

Antal samma: 2

UPPGIFT 3 – KLOCKRADION

Sture har lärt sig se vad klockan är genom att uppskatta ljusnivån i rummet. Ljuset kommer från hans klockradio, vars display alltid visar fyra siffror (t.ex. följer 00:00 efter 23:59) och varje siffra består av ett antal tända streckformade lampor. Siffrorna ser ut så här:



Det Sture gör är att han utifrån ljusnivån kan avgöra det totala antalet tända streck på displayen. Men det finns många tidpunkter som ger samma antal streck. Därför måste han ofta låta tiden gå en stund och göra upprepade mätningar innan han kan vara helt säker på vad klockan är.

Skriv ett program som, givet en serie mätningar (varje minut) av hur många strecklampor som totalt är tända, skriver ut vad klockan är (vid senaste mätningen), så fort detta är entydigt bestämt.

Observera att programmet inte ska vänta på någon signal om att indatan är slut utan skriva ut svaret så fort det går att avgöra tidpunkten.

Körningsexempel 1:

Ljusnivå ? 16
Ljusnivå ? 15
Ljusnivå ? 16
Ljusnivå ? 12

Klockan är 17:51

Körningsexempel 2:

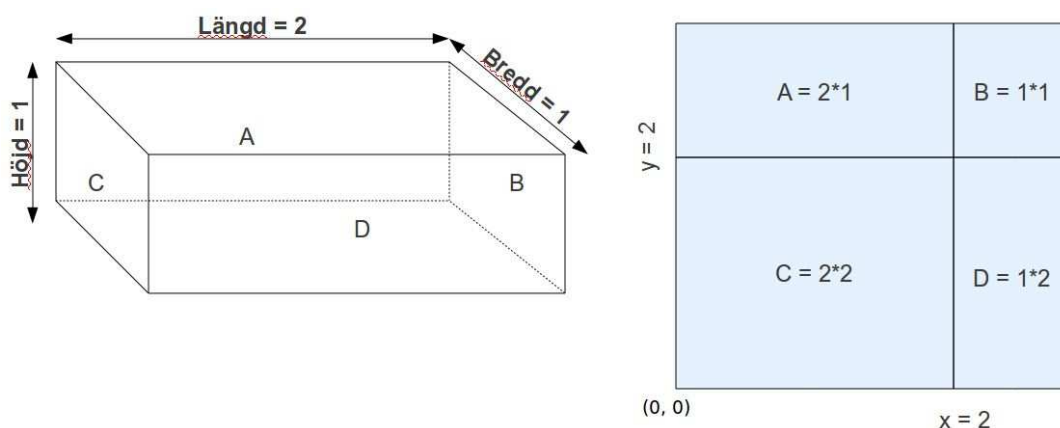
Ljusnivå ? 18
Ljusnivå ? 22
Ljusnivå ? 18

Klockan är 22:01

UPPGIFT 4 – VINDSKYDD

Några snickare ska bygga ett hus. Snickarna bygger först en grund med längden L och bredden B och bygger sedan på inifrån, men för att hålla sig krya medan de arbetar så ska de sätta upp vindskydd. Vindskydden ska stå utmed grundens fyra kanter (d.v.s. där husets väggar sedan kommer) och de ska också ha en viss höjd H .

De fyra vindskyddsväggarna skapas genom att skära upp ett platt rektangulärt plaststycke med dimensionerna $M \times N$ i fyra mindre bitar. Vi tänker oss att originalplasten ligger på en grid med ett hörn i origo. Du skär alltid en gång parallellt med x -axeln och en gång parallellt med y -axeln, och du kan endast skära vid heltalskoordinater. Din uppgift är att bestämma på hur många olika godkända sätt man kan skära upp plasten. För att en skärning ska vara godkänd måste bitarna kunna täcka varsin vägg. Det gör ingenting om bitarna är för stora, men de får inte vara för små.



FIGUR 1. Figuren visar ett av sätten att skära plasten i körningsexempel 1. Observera att alla andra sätt att skära plasten (t.ex. vid $x=1$ och $y=1$) skapar en likadan uppsättning bitar, men de ska ändå räknas som olika skärningar.

Programmet ska fråga efter fem heltal: husets längd, L , och bredd, B , (där $L \geq B$), vindskyddens höjd H , samt plaststyckets dimensioner M och N . Samtliga indatavärden är mellan 1 och 20, inklusive. Programmet ska skriva ut antalet olika sätt att skära plasten på, med avseende på x - och y -koordinaterna.

Körningsexempel 1:

L ? 2
B ? 1
H ? 1
M ? 3
N ? 3

Antal sätt: 4

Körningsexempel 2:

L ? 4
B ? 2
H ? 2
M ? 10
N ? 5

Antal sätt: 14

UPPGIFT 5 – HOPPLEKEN

Barnen i Bullerbyn gillar att leka lekar. Deras favoritlek är *hoppleken* som går till så här: Man ritar upp en 1-dimensionell bana med N st rutor. Det är ett känt antal barn B som leker. Antalet barn får inte vara fler än antalet rutor. Det börjar med att barnen fördelar sig slumpmässigt över rutorna (de får inte ställa sig på samma ruta). Sedan börjar hoppandet. Vid varje hopp kommer samtliga barn att hoppa ett skutt till en ruta bredvid sin egen ruta (de får ej stå stilla). Om några barn hamnar på samma ruta är dessa ute ur leken och lämnar banan. Efter varje hopp kommer man också att ta bort rutan längst till höger så att banan minskar i storlek. Barnen hoppar aldrig utanför banan eller till den ruta som kommer tas bort efter hoppet, men i övrigt hoppar de slumpmässigt åt ena eller andra hållet. Leken är slut efter $N - 2$ hopp, då endast två rutor återstår. De barn som eventuellt då finns kvar i leken vinner. Det kan alltså finnas 0, 1 eller 2 vinnare.

Britta som är överintelligent för sin ålder tycker leken är fänig och vill inte vara med. Hon bestämmer sig istället för att räkna ut det genomsnittliga vinnarantalet. Givet antalet rutor och antalet deltagare, hjälp Britta att besvara hur många vinnare som kommer att finnas i genomsnitt. Tänk på att barnen från början kan ställa sig på alla möjliga sätt.

Programmet ska fråga efter antalet rutor N , samt antalet barn B , där $3 \leq N \leq 20$ och $2 \leq B \leq N$. Programmet ska skriva ut det genomsnittliga antalet vinnare, alltså ett flyttal. Svaret ska anges med minst 4 decimaler.

Körningsexempel 1:

Antal rutor ? 7
Antal barn ? 4

Svar: 0.91429

Körningsexempel 2:

Antal rutor ? 4
Antal barn ? 2

Svar: 1.33333

UPPGIFT 6 – CASTLEVANIA

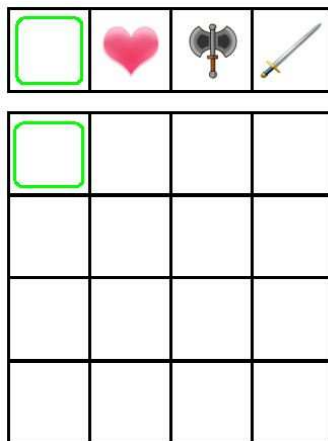
Du spelar Castlevania IV, men du dör hela tiden. För att kunna fortsätta där du dog måste du skriva in en kod varje gång du dött. Det tar tid att skriva in koden, därför bestämmer du dig för att skriva ett program som beräknar det snabbaste sättet att skriva in koden.

Koden i Castlevania skrivs in genom att fylla ett från början blankt 4x4 rutnät med figurer. I en helt separat "räls" med fyra rutor (överst i figuren nedan) väljer man den figur som för tillfället är aktiv, d.v.s. som kan sättas in i rutnätet.

Det finns sålunda tre sorters kommandon:

- Flytta rut-markören i rutnätet ett steg. Markören kan flyttas i alla fyra riktningar. När den går över någon av kanterna går den över till andra sidan.
- Flytta figur-markören i rälsen ett steg. Markören kan flyttas åt höger eller vänster, även denna går över till andra sidan då den går över kanten.
- Placera in den aktiva figuren i rutnätet (på rut-markörens position).

Det finns fyra figurer. De ligger i följande fördefinierade ordning i rälsen: den blanka figuren (), hjärtat (H), yxan (Y) och svärdet (S). I startläget står rut-markören längst upp till vänster och figur-markören längst till vänster (se figur nedan). Samtliga rutor är blanka.



FIGUR 2. Startläget när man ska börja skriva in koden. Översta raden är figur-markörens räls, nedanför finns 4x4 rutnätet där man ska fylla i koden. Notera de två rundade fyrkanterna som representerar markörerna.

Att för hand fylla i figuren med ovanstående regler är visserligen lätt, men du ska skriva ett program som gör det på så få kommandon som möjligt! Programmet ska fråga efter fyra rader med fyra tecken på varje rad: den kod som ska ifyllas i rutnätet. De möjliga tecknen är: `_`, `H`, `Y`, `S`. Programmet ska skriva ut det minsta antalet kommandon som behövs för att fylla i koden.

Körningsexempel 1:

Rad 1? ____
Rad 2? ____
Rad 3? ____
Rad 4? ____S

Minsta antal kommandon: 4

Kommentar: I detta testexempel ska båda markörerna gå över kanten i de möjliga dimensionerna (2 steg för rut- och 1 steg för figurmarkör), samt ett sista kommando för utsättning av svärdet.

Körningsexempel 2:

Rad 1? _H__
Rad 2? _Y__
Rad 3? _Y__
Rad 4? _SH_

Minsta antal kommandon: 15