

Programmeringsolympiaden 2016

TÄVLINGSREGLER FÖR SKOLKVALET

- Tävligen äger rum på av skolan bestämt datum under **fyra timmar** effektiv tid. Eleven ska i förväg komma överens med läraren om att använda egen dator eller en som skolan tillhandahåller. I vilket fall som helst måste eleven befinna sig i av läraren bestämd lokal på skolan.
- Tävligen består av fem uppgifter som vardera ska lösas genom ett datorprogram i valfritt programmeringsspråk.
- Dina lösningar kommer att testköras med förpreparerade indata. Om inget annat anges, testas varje uppgift med 5 testfall, som vardera ger 1 poäng om ditt program skriver ut korrekt svar inom en exekveringstidsgräns av **5 sekunder**.
- Det är ofta olika begränsningar på de olika testfallen, t.ex. storleken på indata eller andra inskränkningar. Detta anges i uppgiften. **Observera att det kan vara helt olika svårighetsgrad på en uppgift beroende på dessa skillnader. Det kan därför vara lättare att få delpoäng på en uppgift som verkar svår än att få full poäng på en uppgift som verkar lättare.** Informationen om delpoäng är därför extremt viktig för att planera sin tävling.
- Ingen test av indata behöver göras, den följer specifikationerna i uppgiften.
- Rättningen utförs på samma eller likvärdig dator. Ändringar i källkoden tillåts ej efter tävlingen. Om programmet inte kan kompileras ges 0 p. på uppgiften.
- Om något av följande inträffar ger det *testfallet* 0 poäng, men programmet fortsätter testas med övriga testfall.
 - Exekveringstiden överstiger 5 sekunder
 - Exekveringsfel (run time error)
 - Fel svar
- Deltagandet är individuellt vilket bland annat innebär att inget utbyte av idéer eller filer får ske under tävlingen.
- Hjälpmedel: Valfritt skriftligt material, material som finns installerat på datorn samt material som finns tillgängligt på internet. Det är *inte* tillåtet att aktivt kommunicera på internet (t.ex. chatta eller ställa frågor till ett forum) utan endast att söka efter information. 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...uppg5 med passande filtillägg. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

Kanske ses vi på den fullspäckade finalhelgen på Spotifys huvudkontor i februari, där landslagsplatser till tävlingar i Finland och Ryssland står på spel.

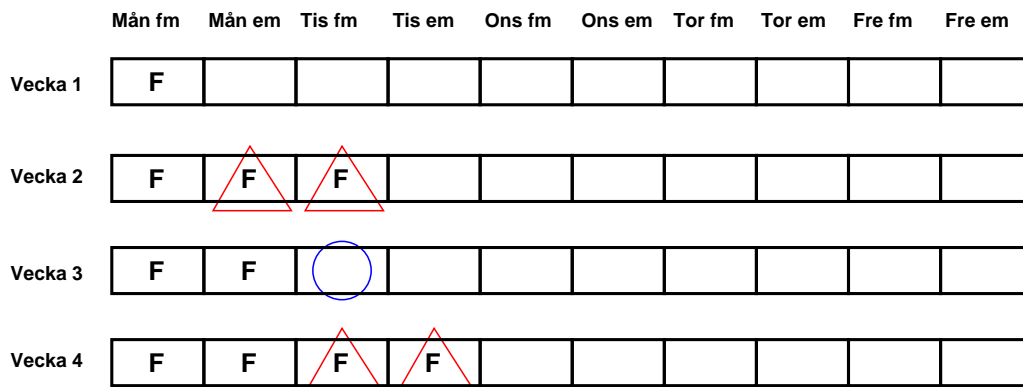
Lycka till!

UPPGIFT 1 – FÖRVIRRAD FÖRELÄSARE

Bjarki undervisar på en kurs på universitetet, men är inte särskilt organiserad av sig. Särskilt förvirrad blir han av att antalet föreläsningar varierar från vecka till vecka.

Första veckan håller Bjarki sina A schemalagda föreläsningar. Men varje vecka utom den första kommer han att förutsätta att schemat är likadant som föregående vecka. Andra veckan håller han alltså exakt A föreläsningar igen. Därför kan det ibland hända att Bjarki håller lektion inför tomt klassrum och ibland att han inte dyker upp när han ska. I slutet av veckan får han dock ett argt brev av sin chef med vilka tider han skulle hållit föreläsningar och kommer istället att använda dessa tider veckan därpå.

Skriv ett program som, givet antalet schemalagda föreläsningar varje vecka, skriver ut antalet föreläsningar Bjarki håller inför tomma klassrum samt antalet föreläsningar Bjarki missar. Det är aldrig mer än 10 föreläsningar under en vecka och tiderna fylls alltid på från början av veckan utan luckor (se figuren). Kursen pågår högst 9 veckor.



FIGUR 1. Schemat i det första exemplet. F markerar schemalagda föreläsningar. En blå cirkel markerar att Bjarki håller lektionen inför tomt klassrum och en röd triangel markerar att han inte dyker upp.

Körningsexempel 1

Antal veckor ? 4
 Vecka 1 ? 1
 Vecka 2 ? 3
 Vecka 3 ? 2
 Vecka 4 ? 4

Tomma: 1
 Missade: 4

Körningsexempel 2

Antal veckor ? 5
 Vecka 1 ? 4
 Vecka 2 ? 1
 Vecka 3 ? 5
 Vecka 4 ? 3
 Vecka 5 ? 10

Tomma: 5
 Missade: 11

Körningsexempel 3

Antal veckor ? 2
 Vecka 1 ? 10
 Vecka 2 ? 10

Tomma: 0
 Missade: 0

UPPGIFT 2 – SIFFERSUMMA

Givet ett heltal N , bestäm det minsta talet som är större än N och har samma siffersumma som N .

Siffersumman av ett heltal är summan av dess siffror. Till exempel har 9550 siffersumman $9 + 5 + 5 + 0 = 19$ och 999 har siffersumman $9 + 9 + 9 = 27$.

Poängsättning

För testfall värt 1 poäng gäller att $1 \leq N \leq 10$

För testfall värt 1 poäng gäller att $1 \leq N \leq 100$

För testfall värda 2 poäng gäller att $1 \leq N < 10\,000$

För testfall värt 1 poäng gäller att $1 \leq N < 10^{20}$

Fyra körningsexempel

$N ? 8$

Svar: 17

$N ? 92$

Svar: 119

$N ? 200$

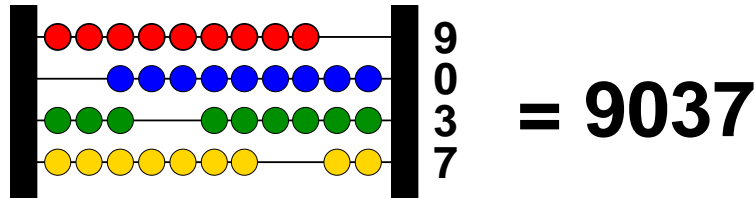
Svar: 1001

$N ? 9550$

Svar: 9604

UPPGIFT 3 – KULRAMEN

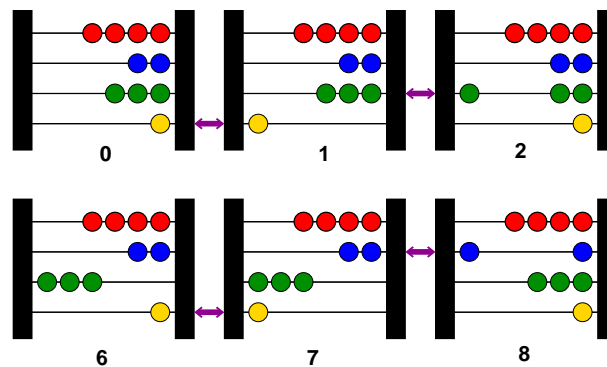
Lille Simon har fått en kulram i present. Kulramen har R rader och i varje rad fanns från början 9 kulor, så att man kunde representera R -siffriga decimaltal – en siffra på varje rad. Om en rad hade X kulor på vänstra sidan, sedan ett mellanrum och övriga kulor på höger sida representerade raden siffran X .



FIGUR 2. Så här kunde Simons kulram (med $R = 4$) se ut innan han började äta upp kulorna. Då var det lätt att översätta ställningen till ett decimaltal.

Tyvärr tyckte Simon att kulorna på ramen såg väldigt smaskiga ut, och åt helt enkelt upp några kulor. Det finns dock minst en kula kvar på varje rad.

Simon lärde sig snabbt att räkna på sin nya kulram. Han representerar talet där alla kulor är på högersidan som talet 0, och adderar sedan 1 precis som han skulle ha gjort på en vanlig kulram, genom att flytta en kula från höger till vänster på den nedersta raden som har någon kula kvar på höger sida (låt oss kalla den F -raden) samt flytta alla kulor på raderna nedanför F -raden till höger sida (om inte F -raden är den nedersta raden). Om 1 adderas när alla kulorna på *alla* rader redan är på vänster sida (så att det inte finns någon F -rad) så flyttas alla kulorna till höger så att resultatet blir 0.



FIGUR 3. Fyra exempel på hur Simon adderar 1 på den kulram som återfinns i de två första exemplen. Dubbelpilen markerar F -raden.

Simon håller på att räkna sandkornen i sin sandlåda och skulle behöva hjälp att skriva ett program som, givet ett visst utgångsläge på kulramen, räknar ut hur kulramen ser ut när han N gånger har adderat 1. Programmet ska fråga efter antalet rader (R), antalet kulor till vänster respektive höger på varje rad (uppifrån och ned), samt slutligen talet N . Programmet ska skriva ut R rader med två tal på varje rad: antalet kulor till vänster respektive höger på varje rad efter additionerna.

Poängsättning

För testfall värt 1 poäng gäller att $R = 4$, $N \leq 100$, och det finns 2 kulor på varje rad.

För testfall värda ytterligare 2 poäng gäller att $R = 4$ och $N \leq 10,000$.

För testfall värda 2 poäng gäller att $R \leq 12$ och $N \leq 10^{12}$.

Körningsexempel 1

Antal rader ? 4
Rad 1, vänster ? 0
Rad 1, höger ? 4
Rad 2, vänster ? 0
Rad 2, höger ? 2
Rad 3, vänster ? 1
Rad 3, höger ? 2
Rad 4, vänster ? 0
Rad 4, höger ? 1
Talet N ? 6

0 4
1 1
0 3
0 1

Här går vi från talet 2
till talet 8 i figuren ovan.

Körningsexempel 2

Antal rader ? 4
Rad 1, vänster ? 2
Rad 1, höger ? 2
Rad 2, vänster ? 2
Rad 2, höger ? 0
Rad 3, vänster ? 2
Rad 3, höger ? 1
Rad 4, vänster ? 1
Rad 4, höger ? 0
Talet N ? 85

1 3
1 1
1 2
0 1

Här passerar vi talet 0 på vägen.

Körningsexempel 3

Antal rader ? 4
Rad 1, vänster ? 1
Rad 1, höger ? 1
Rad 2, vänster ? 0
Rad 2, höger ? 2
Rad 3, vänster ? 2
Rad 3, höger ? 0
Rad 4, vänster ? 1
Rad 4, höger ? 1
Talet N ? 37

2 0
1 1
2 0
2 0

Här är det exakt
2 kulor på varje rad.

Körningsexempel 4

Antal rader ? 10
Rad 1, vänster ? 4
Rad 1, höger ? 5
Rad 2, vänster ? 7
Rad 2, höger ? 2
Rad 3, vänster ? 8
Rad 3, höger ? 0
Rad 4, vänster ? 6
Rad 4, höger ? 3
Rad 5, vänster ? 3
Rad 5, höger ? 5
Rad 6, vänster ? 4
Rad 6, höger ? 4
Rad 7, vänster ? 1
Rad 7, höger ? 8
Rad 8, vänster ? 0
Rad 8, höger ? 9
Rad 9, vänster ? 7
Rad 9, höger ? 1
Rad 10, vänster ? 2
Rad 10, höger ? 6
Talet N ? 9876543210

1 8
5 4
1 7
9 0
7 1
1 7
4 5
3 6
0 8
2 6

UPPGIFT 4 – CHERIMOYOR

Farah älskar den exotiska cherimoya-frukten. Den säljs bara en dag om året i Sverige, men Farah har givetvis passat på att köpa några cherimoyor just denna dag.

Cherimoyorna kan vara olika mogna när hon köper dem. Mer precist är varje cherimoyafrukt ätmogen under totalt tre dagar. Vi säger att frukten *blir ätmogen* en viss dag. Före dess kan man inte äta den och efter de tre dagarna måste den slängas.

Farah vill få ut så mycket som möjligt utav cherimoyasäsongen. Hon vill maximera *njutningen*, som räknas ut såhär: För en given dag får hon 10 njutningspoäng för den första cherimoyan, därefter 9 för den andra, 8 för den tredje o.s.v. Hon orkar aldrig äta mer än 10 cherimoyor på en dag.

Skriv ett program som räknar ut hur många njutningspoäng Farah som mest kan få ut under årets cherimoyasäsong. Programmet ska fråga efter antalet dagar N (högst 15) och hur många cherimoyor som blir ätmogna varje dag (högst 30). Det är alltså under $N+2$ dagar som det är aktuellt att äta cherimoyor. Programmet ska skriva ut hur många njutningspoäng Farah som mest kan få med den bästa ätstrategin.

Poängsättning

För testfall värda 3 poäng kommer N vara högst 5.

För full poäng så ska ditt program klara N upp till 15.

Körningsexempel 1

Antal dagar ? 3
 Mognar dag 1 ? 18
 Mognar dag 2 ? 0
 Mognar dag 3 ? 2
 Svar: 155

Körningsexempel 2

Antal dagar ? 8
 Mognar dag 1 ? 3
 Mognar dag 2 ? 0
 Mognar dag 3 ? 1
 Mognar dag 4 ? 2
 Mognar dag 5 ? 0
 Mognar dag 6 ? 0
 Mognar dag 7 ? 3
 Mognar dag 8 ? 6
 Svar: 144

Körningsexempel 3

Antal dagar ? 2
 Mognar dag 1 ? 30
 Mognar dag 2 ? 30
 Svar: 220

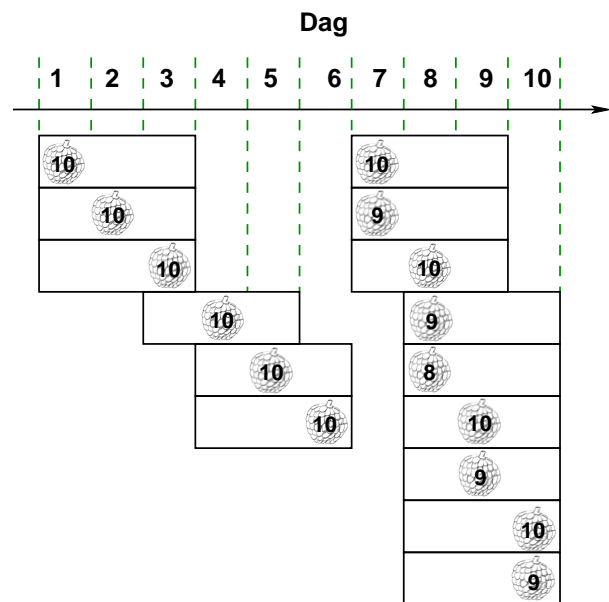
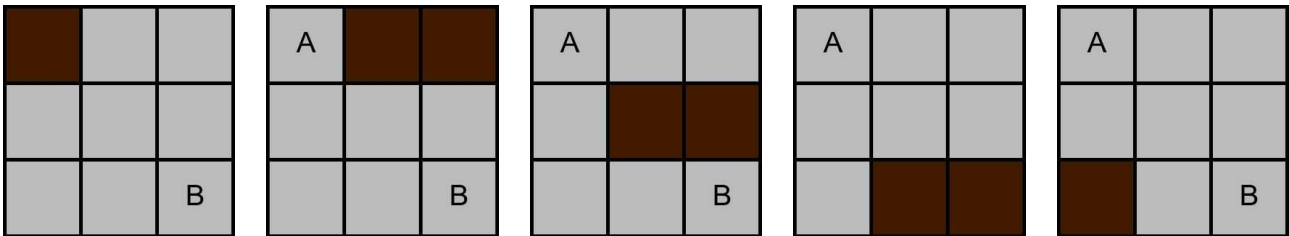


Illustration av det andra exemplet. Varje ruta symboliserar den tredagersperiod då en viss cherimoya kan ätas. Själva frukten är inritad på den dagen som Farah äter den och talet på frukten visar njutningspoängen den ger upphov till (det finns andra lösningar som ger samma maximala summa).

UPPGIFT 5 – RÄTBLOCKET

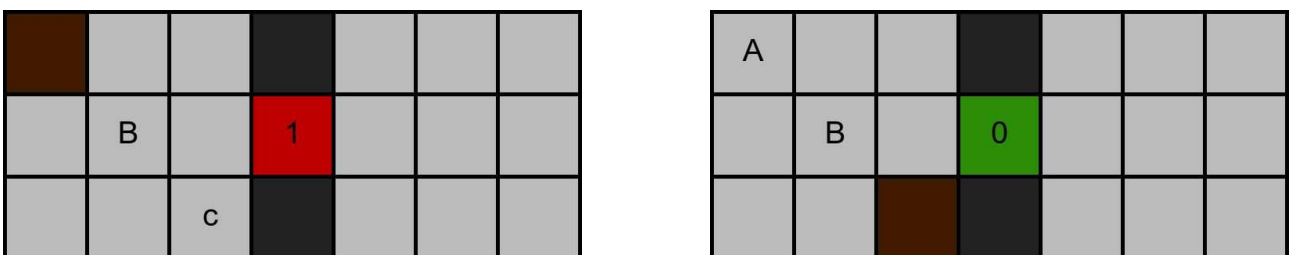
Rätblocket är ett mobilspel som snabbt har blivit otroligt populärt. Spelet går ut på att man genom att vinkla mobilen åt olika håll ska få ett rätblock att rulla från cell A till cell B på en bana. Rätblocket har storlek 1×2 och banan är ett tvådimensionellt bräde med celler av storlek 1×1 . Vissa celler är fria, vilket betyder att rätblocket kan röra sig fritt på dem, medan vissa celler är blockerade. Det är inte möjligt att röra sig utanför banan.

Beroende på om rätblocket står upp eller ligger ner så ockuperar det 1 eller 2 celler. I början av spelet så står blocket upp i cellen markerad med ett A och målet är att flytta det så att det står i cellen markerat med ett B (det räcker alltså inte med att halva blocket ligger på cell B). En förflyttning av blocket kan göras i 4 olika riktningar - upp, ner, höger eller vänster - och en förflyttning gör att blocket rullar över i den givna riktningen (givet att det inte är en blockerad cell i vägen eller banan tar slut). För att tydligare visa hur en förflyttning går till så är här början på en lösning till exempelfall 1:



FIGUR 4. Halva lösningen till det första exempel-fallet. Rätblocket flyttas först ett steg till höger, sedan två steg nedåt och till sist ett steg åt vänster.

På mer avancerade banor så förekommer även så kallade *modulo-celler*. Dessa är celler som kan befinna sig i två olika tillstånd, 0 eller 1. I tillstånd 0 så beter sig en modulo-cell precis som en fri cell som rätblocket kan röra sig på. I tillstånd 1 så är dock modulo-celler upphöjda och beter sig som blockerade celler. Det kan även finnas växel-celler (markerade med ett c) utplacerade på banan. När rätblocket ställs på en växel-cell så ändras tillstånden på alla modulo-celler (0 blir till 1 och vice versa). Observera att även på en växelcell måste blocket stå upp för att det ska räknas.



FIGUR 5. Det tredje exempel-fallet, före och efter att växeln har aktiverats.

Banorna är designade så att det alltid finns en giltig lösning. Din uppgift är att beräkna det minsta antal förflyttningar som behövs för att lösa banan, d.v.s. flytta rätblocket från cell A till cell B .

Indata

På denna uppgift bör du skriva programmet så att hela indatan kan kopieras–klistras in vid testningen. Programmet behöver därför inte föra någon dialog med användaren. Om du är osäker kan du välja att göra en dialog istället.

Du kommer först att få en rad med två heltal N och M . Detta betyder att banan har N rader och M kolumner ($3 \leq N, M \leq 7$)

De följande N raderna har M tecken var och beskriver hur banan ser ut. Följande tecken kan förekomma:

- A eller B : beskriver banans start- respektive slut-positioner.
- $.$ (en punkt): beskriver en fri cell.
- $\#$: beskriver en blockerad cell.
- 0 eller 1 : beskriver en modulo-cell med givet start-tillstånd.
- c : beskriver en växel-cell

Det är garanterat att varje bana innehåller precis ett A och ett B .

Poängsättning

För testfall värda 2 poäng så har banorna inga modulo-celler.

För de resterande 3 poängen så behöver din lösning även kunna hantera modulo-celler.

Körningsexempel 1

```
3 3
A..
...
..B
```

Svar: 8

Körningsexempel 2

```
3 5
A#...
.....
...#B
```

Svar: 14

Körningsexempel 3

```
3 7
A..#...
.B.1...
..c#...
```

Svar: 27