

Programmeringsolympiaden 2014

TÄVLINGSREGLER FÖR SKOLKVALET

- Tävligen äger rum på av skolan bestämt datum under sex 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 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 2 februari. För mer information, se www.progolymp.se
- Samtliga uppgifter leder fram till program vars exekveringstid normalt bör understiga 1 sekund. Om programmets exekveringstid överstiger 5 sekunder bedöms programmet för det testexemplet som felaktigt.
- Rättningen utförs på samma eller likvärdig dator. Om fel uppstår vid kompilering bedöms programmet 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.
- 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...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 Taiwan i juli. Kanske blir du en av dem som representerar Sverige där.

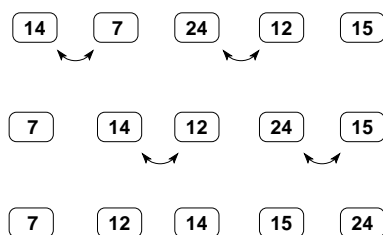
Lycka till!

UPPGIFT 1 – SORTERA SPELLISTAN

OBS! Denna uppgift kan alternativt lösas via *Kattis*. Se vidare på www.progolymp.se

Du har en spellista med N låtar ($1 \leq N \leq 10$), där alla har olika längd (ett heltal mellan 1 och 1000). Låtarna ligger från början i en given ordning. Du vill nu sortera listan så att de kortaste låtarna kommer först och de längsta låtarna kommer sist.

Vad är det minsta antalet platsbyten du behöver göra för att få listan sorterad? Vid ett platsbyte väljer du två intilliggande låtar och byter plats på dem.



FIGUR 1. Översta raden visar låtarnas startordning i första exemplet. Pilarna visar platsbytena som behöver göras för att göra spellistan sorterad (understa raden)

Körningsexempel 1

Antal låtar ? 5
 Längd 1 ? 14
 Längd 2 ? 7
 Längd 3 ? 24
 Längd 4 ? 12
 Längd 5 ? 15

Antal platsbyten: 4

Körningsexempel 2

Antal låtar ? 7
 Längd 1 ? 11
 Längd 2 ? 9
 Längd 3 ? 5
 Längd 4 ? 3
 Längd 5 ? 7
 Längd 6 ? 2
 Längd 7 ? 10

Antal platsbyten: 14

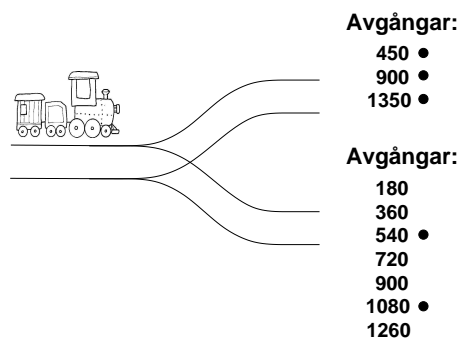
UPPGIFT 2 – TÅGVÄXELN

Växelholm är en väldigt liten stad med en enda byggnad – Växelholms tågstation – och en enda invånare, nämligen tågstationens föreståndare, Lokas. Lokas jobb går ut på att operera stationens manuella tågväxel, så att tågen på de två pendeltågslinjerna som passerar genom staden åker åt rätt håll. Tågen går periodiskt med m respektive n minuters mellanrum på de två linjerna, med första avgång m respektive n minuter efter midnatt. Alla tåg åker alltså ut från stationen åt samma håll men delas sedan upp på två olika spår med hjälp av en växel.

Nu har JS, Järnvägarnas Stat, bestämt att Lokas ska få lön baserat på hur många gånger han ändrar växeln under ett dygn, närmare bestämt fr.o.m. 0:00 t.o.m. 23:59 (d.v.s. mellan 0 och 1439 minuter efter midnatt). Lokas ska ändra växeln enligt reglerna:

- (1) Om ett tåg avgår och växeln är fel inställd, ska Lokas ändra växeln till rätt spår.
- (2) Om två tåg avgår samma minut, så avgår först det tåg som växeln är inställd för, och sedan ska Lokas ändra växeln till det andra tågets spår.
- (3) I början av dygnet är växeln inställd på spåret för det tåg som avgår först.

Skriv ett program som beräknar hur många gånger som Lokas måste ändra växeln under ett helt dygn. Programmet ska fråga efter de två heltalen m och n , där $m \neq n$ och $1 \leq m, n \leq 1439$, som är perioden för tågens avgång, angiven i minuter.



FIGUR 2. Avgångstiderna (i minuter efter midnatt) från Växelholm på vardera pendeltågslinjen i det första exemplet. En punkt efter tiden innebär att Lokas måste ställa om växeln innan denna avgång.

Körningsexempel 1

Talet m ? 450
 Talet n ? 180
 Antal växlingar: 5

Körningsexempel 2

Talet m ? 500
 Talet n ? 1000
 Antal växlingar: 1

Körningsexempel 3

Talet m ? 719
 Talet n ? 720
 Antal växlingar: 2

Körningsexempel 4

Talet m ? 7
 Talet n ? 4
 Antal växlingar: 410

UPPGIFT 3 – KRYPTO

Ett av de mest komplicerade *manuella* kryptona som användes i fält (t.ex. under första världskriget) kallas *dubbelt kolumnvis transpositionskrypto*, där ordet transpositionskrypto betyder att bokstäverna inte ändras utan bara byter plats.

För att använda denna metod behövs en hemlig *nyckel* som bara avsändaren och mottagaren har tillgång till. I det här fallet är nyckeln ett ord där ingen bokstav får förekomma flera gånger. Nyckelns längd är viktig, vi kan kalla detta tal för N . Låt oss till exempel anta att nyckeln är KODA, d.v.s. $N = 4$.

När man vill kryptera en text (alltid utan mellanslag) skriver man den med N tecken på varje rad (den sista raden kan bli kortare om det inte går jämnt upp). Sedan får man den nya texten genom att läsa kolumnvis istället, men man läser kolumnerna i en speciell ordning som avgörs av nyckeln. Om man låter bokstäverna i nyckeln *namnge* kolumnerna så läser man dem i bokstavsordning. I vårt exempel börjar man alltså med A, d.v.s. den fjärde kolumnen, därefter D, d.v.s. den tredje kolumnen, därefter K, d.v.s. den först kolumnen, och slutligen O, den andra kolumnen.

Detta visas lättast med ett exempel. Antag att vi vill kryptera texten VISESIFINALEN med nyckeln KODA. Vi skriver då bokstäverna som figuren till höger visar, och läser kolumnvis: EIESFLVSNNIIA

K	O	D	A
V	I	S	E
S	I	F	I
N	A	L	E
N			

Efter denna procedur är det ofta ganska enkelt att knäcka koden genom att gissa en nyckellängd och försöka känna igen de omkastade orden (anagram) som bildats. Därför upprepar man samma procedur en gång till för att göra det svårare, därav namnet *dubbelt kolumnvis transpositionskrypto*. I vårt exempel får vi den dubbelkrypterade texten SSIEVIEFNAILN.

Skriv ett program som frågar efter nyckeln (högst 10 tecken, alla olika och valda bland A-Z) samt den dubbelkrypterade texten (högst 20 tecken valda bland A-Z) och skriver ut det ursprungliga meddelandet.

Körningsexempel 1

Nyckel ? KODA
 Krypterad ? SSIEVIEFNAILN
 Ursprunglig: VISESIFINALEN

Körningsexempel 3

Nyckel ? X
 Krypterad ? DUMNYCKELELLERHUR
 Ursprunglig: DUMNYCKELELLERHUR

Körningsexempel 2

Nyckel ? HEMLIGT
 Krypterad ? RAAVNBAIEVDANL
 Ursprunglig: ANVANDVARIABEL

UPPGIFT 4 – LEKSAKSROBOTEN

Johanna har fått en leksaksrobot i present. Roboten är ganska speciell, så interaktion med den sker via ett text-interface där man skriver in instruktioner som roboten sedan följer. Det finns tre typer av instruktioner:

- V Detta gör att roboten roterar 90 grader åt vänster
- H Detta gör att roboten roterar 90 grader åt höger
- X, där X är ett heltal och $0 \leq X \leq 1000$. Detta gör att roboten rör sig X decimeter framåt (d.v.s. i den riktning den för tillfället tittar).

Efter att ha lekt med roboten i många timmar inser Johanna att den endast förflyttat sig några decimeter från ursprungspositionen. Om hon lekt mer effektivt kanske hon kunnat sluta i den positionen mycket tidigare och haft tid att göra något vettigt istället.

Din uppgift är att utifrån en lista av instruktioner (högst 15 stycken) skapa en ny lista som tar roboten till samma slutposition (men inte nödvändigtvis tittande i samma riktning). Den nya listan måste dock ha *minimalt antal instruktioner*. Om det finns flera möjliga svar så kan vilket som helst ges.

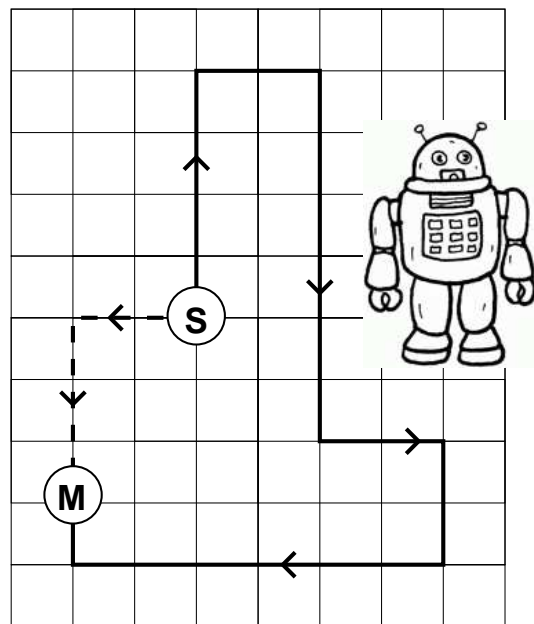
Körningsexempel 1

Antal instruktioner ? 13

- Instruktion 1 ? 4
- Instruktion 2 ? H
- Instruktion 3 ? 2
- Instruktion 4 ? H
- Instruktion 5 ? 6
- Instruktion 6 ? V
- Instruktion 7 ? 2
- Instruktion 8 ? H
- Instruktion 9 ? 2
- Instruktion 10 ? H
- Instruktion 11 ? 6
- Instruktion 12 ? H
- Instruktion 13 ? 1

Ny lista:

- V
- 2
- V
- 3



Den tjocka heldragna linjen visar robotens väg från startpositionen (S, tittande uppåt) till målpositionen (M) i första exemplet. Den streckade linjen visar en alternativ väg som kräver minimalt antal instruktioner, fyra stycken.

Körningsexempel 2

Antal instruktioner ? 5

Instruktion 1 ? V

Instruktion 2 ? 5

Instruktion 3 ? V

Instruktion 4 ? V

Instruktion 5 ? 6

Ny lista:

H

1

Körningsexempel 3

Antal instruktioner ? 5

Instruktion 1 ? H

Instruktion 2 ? 10

Instruktion 3 ? V

Instruktion 4 ? 3

Instruktion 5 ? V

Ny lista:

3

H

10

UPPGIFT 5 – IP-ADRESSER

En IPv4-adress består av fyra heltal mellan 0 och 255 (som inte får ha några inledande nollor), separerade av punkter. T.ex. är 1.0.3.255 en giltig adress, medan 1.0.03.255, 1.0.3.256 och 1.0.3 inte är giltiga adresser.

Skriv ett program som läser in en sekvens av siffror och räknar ut hur många giltiga IPv4-adresser som kan skapas genom insättning av tre punkter i sekvensen. Sekvensen består av minst 4 och högst 12 siffror.

Körningsexempel 1

Sekvens ? 291841
Antal: 7

Förklarande listor:

2.9.18.41
2.9.184.1
2.91.8.41

Körningsexempel 2

Sekvens ? 00000
Antal: 0

2.91.84.1
29.1.8.41
29.1.84.1
29.18.4.1

Körningsexempel 3

Sekvens ? 255255255255
Antal: 1

255.255.255.255

Körningsexempel 4

Sekvens ? 0000
Antal: 1

0.0.0.0

Körningsexempel 5

Sekvens ? 20142014
Antal: 6

20.14.20.14
20.14.201.4
20.142.0.14
201.4.20.14
201.4.201.4
201.42.0.14

UPPGIFT 6 – STÅSKRIVBORDET

Annas mamma klagar på att hon *sitter* för mycket vid datorn varje dag. Men detta ska Anna minsann ordna. Hon har bestämt sig för att *stå* upp vid datorn istället!

Men datorskärmen står för högt upp! Anna behöver komma upp exakt x centimeter av ergonomiska skäl. Hon har n små plattor till sin hjälp. Plattorna har höjderna $h_1 \dots h_n$. Genom att lägga ett antal av dem under vardera foten, så kan hon komma upp exakt x centimeter. Plattorna under höger fot måste alltså ha sammanlagd höjd x centimeter, och plattorna under vänster fot måste också ha sammanlagd höjd x centimeter. Vad är det *minsta totala antalet* plattor hon behöver?

Programmet ska fråga efter den önskade höjden x och antalet plattor n , där $10 \leq x \leq 100$ och $2 \leq n \leq 20$. Därefter ska programmet fråga efter plattornas höjder $h_1 \dots h_n$, där $1 \leq h_i \leq 100$. För att göra det svårare att gissa rätt svar ska du skriva ut två heltal n_1 och n_2 , antalet plattor hon har under vardera foten. Skriv det minsta talet först.

Det kommer alltid att finnas en lösning. Om det finns flera lösningar (n_1, n_2) med samma *minimala* summa $n_1 + n_2$, kan du välja vilken som helst av dem (men se till att $n_1 \leq n_2$).

Körningsexempel 1

Önskad höjd ? 14
Antal plattor ? 5
Platta 1 ? 10
Platta 2 ? 5
Platta 3 ? 4
Platta 4 ? 9
Platta 5 ? 7

Svar: 2 2

Körningsexempel 2

Önskad höjd ? 100
Antal plattor ? 9
Platta 1 ? 30
Platta 2 ? 33
Platta 3 ? 33
Platta 4 ? 4
Platta 5 ? 50
Platta 6 ? 40
Platta 7 ? 60
Platta 8 ? 51
Platta 9 ? 48

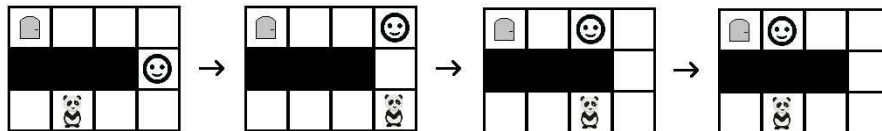
Svar: 2 4

UPPGIFT 7 – GROTTFLYKT

Av okänd anledning håller du på att utforska en grotta och har råkat väcka en arg björn. Björnen är större och snabbare än du, så din enda chans är att överlista den för att ta dig ut. Grottan kan representeras som ett rektangulärt bräde bestående av väggar och en utgång. Du och björnen kan förflytta er i riktningarna upp, ner, höger eller vänster, men kan inte gå igenom väggar.

Varje gång du tagit ett steg, och inte kommit fram till utgången, så kommer björnen svara genom att flytta högst 2 steg för att försöka ta sig så nära dig som möjligt. Om ni någon gång står på samma ruta så är du förlorad. Som tur är så är björnen inte helt medveten om väggar, och använder sig därför av en korkad algoritm för att flytta sig.

Det första björnen gör är att flytta sig mot dig i horisontell riktning (höger eller vänster). Detta gör den tills den fått slut på steg, nått en vägg eller befinner sig i samma kolumn som du. Därefter gör den likadant i vertikal riktning (upp eller ner), tills den fått slut på steg, nått en vägg eller är i samma rad som du. Det är alltså möjligt att björnen förflyttar sig 0, 1 eller 2 steg beroende på var du befinner dig och var det finns väggar.



FIGUR 3. En illustration av det första exemplet.

Givet utseendet på grottan är det din uppgift att skriva ett program som hittar en sekvens av förflyttningar som låter dig fly grottan utan att bli fångad. Varje givet testfall går att lösa och ofta finns det flera möjliga sätt (som t.ex. i körningsexempel 3 nedan).

Programmet ska fråga efter två heltal w och h , grottans storlek, där $1 \leq w \leq 8$ och $1 \leq h \leq 8$. Sedan ska det läsa in h rader med w tecken var, en beskrivning av grottan. En vägg beskrivs med '#', din startposition med 'S', utgången med 'U', björnen med 'B' och övriga rutor med punkt. Programmet ska skriva ut en sträng bestående av tecknen 'U' för upp, 'N' för ner, 'H' för höger och 'V' för vänster, stegen du ska ta för att fly grottan. Ett svar som någon gång leder in i en vägg kommer inte godkännas. Det är alltså omöjligt att stå still en omgång. **Delpoäng:** För delpoäng kommer grottan bestå av högst 25 celler. För full poäng kan banorna ha upp till 64 celler.

Körningsexempel 1

w ? 4
h ? 3
Rad 1 ? U...
Rad 2 ? ###S
Rad 3 ? .B..

Steg: UVVV

Körningsexempel 2

w ? 3
h ? 3
Rad 1 ? .U#
Rad 2 ? ..S
Rad 3 ? B#..

Steg: VU

Körningsexempel 3

w ? 3
h ? 5
Rad 1 ? B.U
Rad 2 ? .#.
Rad 3 ? #..
Rad 4 ? S..
Rad 5 ? ...

Steg: NUHUHU