

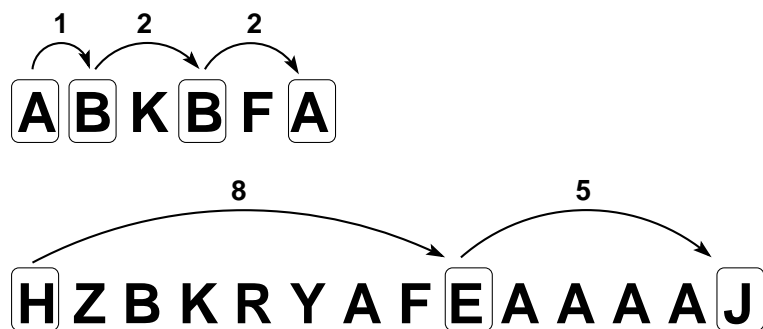
Programmeringsolympiaden 2015

TÄVLINGSREGLER FÖR SKOLKVALET

- Tävlingen 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.
- 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.
- En uppgift, *Plocka äpplen*, är gemensam för skolkvalet och onlinekvalet och kan därför alternativt lösas via vår rättningsdomare Kattis, gärna redan före tävlingsveckan men senast söndag 30 november. För att poängen ska räknas med i skolkvalet krävs att du fyller i ett formulär på 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.

Kanske ses vi på den fullspäckade finalhelgen på Spotifys huvudkontor, där landslagsplatser till tävlingar i Polen och Kazakstan står på spel.

Lycka till!



FIGUR 1. Illustration av dekrypteringen i de två exemplen.

UPPGIFT 1 – GÖMDA ORD

Anna skickar hemliga krypterade meddelanden till Bert. För att kunna läsa meddelandena måste Bert dekryptera dem med följande algoritm:

- Första bokstaven i indata-strängen tas med i utdatasträngen.
- Varje bokstav som man tar med beskriver var i indatasträngen nästa bokstav finns som ska tas med. Ett 'A' betyder att nästa bokstav finns 1 position fram, ett 'B' innebär 2 positioner fram osv.
- När man kommit till den sista bokstaven i indatasträngen så tar man med den bokstaven och är klar. Indatan är sådan att man alltid kommer till den sista bokstaven.

Hjälp Bert genom att skriva ett program som dekrypterar Annas meddelanden.

Programmet ska fråga efter den krypterade strängen (högst 50 tecken, versaler A-Z) och skriva ut klartexten (den avkrypterade strängen). Strängen kommer kunna avkrypteras med ovan beskrivna algoritm utan att man trillar över sista bokstaven.

Körningsexempel 1

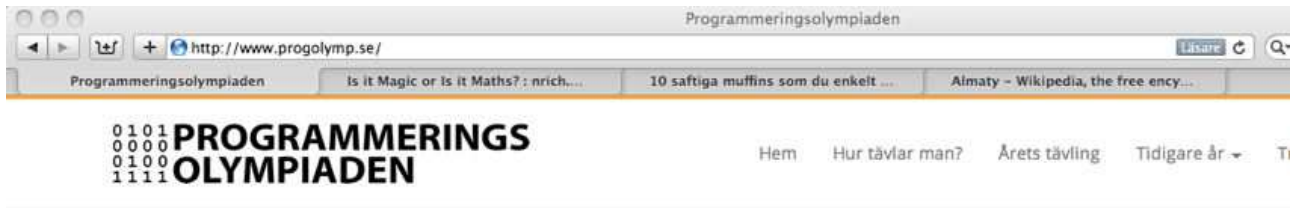
Krypterad ? ABKBFA

Klartext: ABBA

Körningsexempel 2

Krypterad ? HZBKRYAFEAAAAJ

Klartext: HEJ



FIGUR 2. En webbläsare med fyra öppna tabbar.

UPPGIFT 2 – TABBTABBANDE

När man arbetar med en webbläsare så händer det ofta att man har väldigt många tabbar (flikar) öppna samtidigt.

Ett vanligt sätt att navigera mellan dem är att ctrl-tabba för att gå igenom dem i den ordning som de ligger. Det går även att ctrl-shift-tabba för att gå igenom dem i omvänd ordning. Tabbarna kan tänkas ligga cykliskt, så det går att ctrl-tabba från sista till första, och ctrl-shift-tabba från första till sista tabben.

Just nu har du N tabbar öppna, numrerade från 1 till N i den ordning som de ligger. Från början har du tabb 1 markerad. Givet en sekvens som beskriver vilka tabbar som ska användas och i vilken ordning, beräkna hur många gånger du minst måste trycka på tabb-tangenten för att besöka dem?

Programmet ska fråga efter de två heltalen N och M : antalet tabbar respektive antalet gånger du vill växla till att använda en annan tabb. Båda dessa heltal är mellan 1 och 10. Därefter ska programmet läsa in ordningen på de tabbar du vill växla till, d.v.s. M heltal mellan 1 och N . Det förekommer inte att man ska växla till den tabb man redan är på.

Programmet ska skriva ut ett heltal, det minsta antalet gånger tabbknappen måste tryckas på.

Körningsexempel 1

Antal tabbar ? 5
Antal växlingar ? 3
Växla till ? 2
Växla till ? 5
Växla till ? 4

Svar: 4

Körningsexempel 2

Antal tabbar ? 9
Antal växlingar ? 5
Växla till ? 5
Växla till ? 9
Växla till ? 4
Växla till ? 9
Växla till ? 8

Svar: 17

UPPGIFT 3 – MUFFINSPELET

Alf och Beata var två ungdomar som levde för länge, länge sedan, på tiden innan man kunde spendera sina eftermiddagar med programmeringstävlingar. Deras liv var således mycket tråkigare än vad dagens ungdomars liv är. Hur man kan överleva utan datorer, kanske du frågar dig. Svaret är enkelt: man bakar!

Våra två ungdomar älskade att baka muffins, och hade ofta stora högar när de var klara med bakningen. För att inte fylla sina kök med muffins utmanade Beata sin kompis på ett spel varje kväll - *Muffinspelet*.

Muffinspelet spelas av två spelare (i vårt fall, Alf och Beata), samt en stor hög med N muffins. Spelarna turas nu om att göra drag. Ett drag går ut på att spelare A delar upp muffinshögen i två delar (där en av högarna kanske är tom). Motspelaren väljer sedan en av högarna, och äter upp alla muffins i högen. I nästa drag byter spelarna roll, så spelare B delar upp muffinshögen och spelare A äter upp en av högarna. De turas om på detta vis ända tills alla muffins är slut.

Alf börjar med att göra ett drag (dvs att dela upp den stora högen), och Beata börjar med att äta upp en av högarna. Kan du beräkna hur många muffins Alf och Beata kommer äta under spelets gång om de båda spelar så bra som möjligt (alltså vill ha så många muffins som möjligt själva)?

Programmet ska fråga efter antalet muffins N i högen från början. Programmet ska skriva ut två heltal: antalet muffins som Alf kommer äta och antalet muffins som Beata kommer äta om de båda spelar så bra som möjligt.

Poängsättning: För att få hälften av poängen måste du klara testfall där $N \leq 20$. För att få full poäng måste du klara testfall där $N \leq 10\,000$.

Ledning: när man delar en hög med muffins vill man alltid göra det i två högar vars storlekar är så lika som möjligt (se exempelförklaringarna).

Fyra körningsexempel

Antal muffins ? 1
Svar: 0 1

Eftersom det bara finns en muffin är den enda möjliga uppdelningen Alf kan göra en tom hög och en hög med en muffin. Beata kommer då äta upp högen med en muffin.

Antal muffins ? 4
Svar: 1 3

Här kan Alf bara få en muffin. Den första rundan delar han upp muffinshögen i två högar med 2 muffins var. Beata äter upp 2 muffins, och delar sedan upp den kvarvarande högen i 2 högar med 1 muffin var. Alf äter upp en muffin och måste sedan låta Beata få sista muffinen.

Antal muffins ? 8
Svar: 3 5

Antal muffins ? 13
Svar: 4 9

UPPGIFT 4 – KANINHÅL

I den nyss avslutade tävlingen Databävern (www.databavern.se) fick eleverna se ett exempel på djurens märkliga samspel:

En grupp med N bävvar ska gå på promenad i skogen. De går på ett led efter varandra, den ena bävern efter den andra. Men de busiga kaninerna har grävt en massa hål utefter stigen som bävrarna går på.

Hålen är tillräckligt djupa för att ett visst antal bävvar ska falla i dem. När hålet väl är fullt med bävvar kan de bakomvarande bävrarna passera ovanpå bävrarna i hålet, tills slutligen den sista bävern i raden drar upp bävrarna ur hålet, den översta först och den understa sist. Alltså, om vi har fem bävvar (5 4 3 2 1) som vandrar åt höger (nummer 1 går alltså först och nummer 5 sist i ledet), och de kommer till ett hål där tre bävvar får plats (d.v.s. ett hål med djupet 3), så skulle följande hända:

Från början	De tre första bävrarna har fallit i hålet.	De återstående har passerat och börjar dra upp.	Alla bävrarna är uppe ur hålet och på ett led igen.

Tänk dig nu att kaninerna har gjort *tre hål* efter varandra på stigen (vardera med ett djup mellan 1 och $N-1$). Skriv ett program som, givet hur raden ser ut efter att bävrarna passerat alla hålen, beräknar djupet för varje hål.

Programmet ska fråga efter talet N , antalet bävvar, där $2 \leq N \leq 10$. Sedan ska programmet läsa in ordningen på bävrarna när de passerat de tre kaninhålen. Detta görs genom att fråga efter numret på varje bäver, i ordning från vänster till höger. Alla dessa tal är mellan 1 och N , och alla är olika. Från början är ordningen $N, N-1, N-2, \dots, 1$. Observera att de vandrar åt höger, så bäver 1 går först i ledet.

Programmet ska skriva ut tre heltal mellan 1 och $N-1$, djupet på det första, andra respektive tredje kaninhålet. Du kan förutsätta att det finns exakt en lösning.

Körningsexempel 1

Antal bävvar ? 5
 Nummer ? 3
 Nummer ? 4
 Nummer ? 2
 Nummer ? 1
 Nummer ? 5

Hålens djup: 1 1 2

Körningsexempel 2

Antal bävvar ? 7
 Nummer ? 3
 Nummer ? 2
 Nummer ? 1
 Nummer ? 4
 Nummer ? 6
 Nummer ? 7
 Nummer ? 5

Hålens djup: 5 4 4

UPPGIFT 5 – PLOCKA ÄPPLEN

OBS! Denna uppgift kan alternativt lösas via Kattis, fram till 30 november. För att räkna med den i skolvalet måste du då fylla i ett formulär. Se vidare på www.progolymp.se

IOI 2015 avgörs i Almaty, “äpplets fader”. Olga har en äppelodling med två rader träd. I varje rad finns det N träd. Varje träd har ett visst antal mogna äpplen.

Olga börjar besöka trädet i det sydvästra hörnet (det längst till vänster på den undre raden), och plockar alla dess äpplen. När hon är färdig med ett träd går hon till ett av de närmaste träden (åt norr, öster, väster eller söder) och plockar dess äpplen.

Skriv ett program som beräknar hur många äpplen Olga sammanlagt kan plocka om hon hinner besöka högst K träd.

Programmet ska fråga efter antalet träd i varje rad, N , samt antalet träd Olga hinner besöka, K . I alla testfall gäller att $1 \leq N \leq 15$ och $1 \leq K \leq 15$. Sedan ska programmet fråga efter antalet äpplen på var och ett av de N träden i norra raden, och slutligen efter antalet äpplen på var och ett av de N träden i södra raden, i ordning från väster till öster. Antalet äpplen på varje träd är mellan 0 och 1000. Programmet ska skriva ut antalet äpplen Olga hinner plocka.

OBS! Poängsättning: För att få en poäng räcker det att klara testfall där varje träd har lika många äpplen. För två poäng kan det finnas olika antal äpplen på olika träd.

Körningsexempel 1

Antal träd per rad ? 2
 Antal hon hinner besöka ? 2
 Norra raden, träd 1 ? 7
 Norra raden, träd 2 ? 8
 Södra raden, träd 1 ? 6
 Södra raden, träd 2 ? 4
 Svar: 13

Förklaring: Trädet Olga börjar på har 6 äpplen. Trädet åt norr har 7 äpplen, medan trädet åt öster bara har 4 äpplen. Hon plockar därför $6 + 7 = 13$ äpplen.

Körningsexempel 2

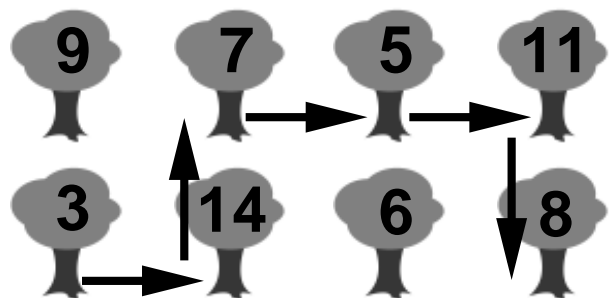
Antal träd per rad ? 3
 Antal hon hinner besöka ? 7
 Norra raden, träd 1 ? 5
 Norra raden, träd 2 ? 5
 Norra raden, träd 3 ? 5
 Södra raden, träd 1 ? 5
 Södra raden, träd 2 ? 5
 Södra raden, träd 3 ? 5
 Svar: 30

Förklaring: Olga skulle hinna besöka 7 träd, men det finns bara 6 träd i odlingen.

Körningsexempel 3

Antal träd per rad ? 4
 Antal hon hinner besöka ? 6
 Norra raden, träd 1 ? 9
 Norra raden, träd 2 ? 7
 Norra raden, träd 3 ? 5
 Norra raden, träd 4 ? 11
 Södra raden, träd 1 ? 3
 Södra raden, träd 2 ? 14
 Södra raden, träd 3 ? 6
 Södra raden, träd 4 ? 8
 Svar: 48

Förklaring: Olgas väg visas nedan.



UPPGIFT 6 – TANKELÄSNING

Ett vanligt magitrick går till på följande vis:

Tänk på ett tal. Subtrahera 1 från talet du tänkte på. Multiplicera resultatet med 3 och lägg till 9. Dela svaret med 3. Addera 5. Subtrahera talet du tänkte på från början. Talet du har kvar är 7.

Självklart är ingen verklig tankeläsning involverad. Om vi kallar det ursprungliga talet för x får vi genom att utföra operationerna uttrycken

$$\begin{aligned}(x) - 1 &= x - 1 \\(x - 1) * 3 &= 3x - 3 \\(3x - 3) + 9 &= 3x + 6 \\(3x + 6) / 3 &= x + 2 \\(x + 2) + 5 &= x + 7 \\(x + 7) - x &= 7\end{aligned}$$

Alltså kommer vi alltid få talet 7 på slutet!

Johan vill imponera på sina kompisar genom att utföra en liknande tankeläsning. Tyvärr är han inte så bra på matte, så han behöver hjälp att ta reda på om hans eget magitrick faktiskt fungerar.

Programmet ska fråga efter ett heltal N ($1 \leq N \leq 10$) – antalet operationer som magitricket består av. Därefter ska det fråga efter var och en av de N operationerna. Varje operation beskrivs med hjälp av två mellanslagsseparerade tecken. Det första tecknet kommer att vara ett av '+', '-', '*', och '/', och beskriver operationen som utförs. Det andra tecknet anger talet operationen utförs med, och kommer att vara antingen en siffra 0-9, eller ett 'x'. I det senare fallet ska du utföra operationen med talet som man ursprungligen tänkte på i stället.

Du kommer aldrig få kommandot "/ x" eller "/ 0".

Om magitricket fungerar, d.v.s. om man alltid får samma tal i slutet oavsett vad man började med, och det talet dessutom är ett heltal, ska programmet skriva ut talet. I annat fall ska det skriva ut "Nej". Observera att det enbart är sluttalet som måste vara ett heltal – tal som uppträder i uträkningen behöver inte vara det.

Körningsexempel 1

Antal operationer ? 6
Operation ? - 1
Operation ? * 3
Operation ? + 9
Operation ? / 3
Operation ? + 5
Operation ? - x

Svar: 7

Körningsexempel 2

Antal operationer ? 6
Operation ? - 1
Operation ? * 3
Operation ? + 9
Operation ? / 2
Operation ? + 5
Operation ? - x

Svar: Nej

Fler exempel finns på nästa sida...

Körningsexempel 3

Antal operationer ? 6
Operation ? + 2
Operation ? - x
Operation ? * x
Operation ? / 2
Operation ? + 3
Operation ? - x

Svar: 3

Körningsexempel 4

Antal operationer ? 5
Operation ? + 7
Operation ? * x
Operation ? * 0
Operation ? * x
Operation ? - 7

Svar: -7

Körningsexempel 5

Antal operationer ? 1
Operation ? * x

Svar: Nej

Körningsexempel 6

Antal operationer ? 4
Operation ? * 3
Operation ? / 3
Operation ? - x
Operation ? + 5

Svar: 5

Körningsexempel 7

Antal operationer ? 3
Operation ? + 1
Operation ? - x
Operation ? / 2

Svar: Nej

UPPGIFT 7 – REGERING

Väljarna i PO-land har röstat och N partier har fått plats i parlamentet, vardera med ett visst antal mandat. Eftersom alla partier tycker ungefär likadant i PO-land (rekursion istället för inflation etc.) funderar talmannen på att lotta ut regeringsmakten. Men då måste hon först veta hur många möjliga majoritetsregeringar det finns.

Skriv ett program som beräknar på hur många sätt man kan bilda regering så att regeringen har majoritet i parlamentet, d.v.s. så att de ingående partierna tillsammans har fler mandat än övriga partier. Regeringen får inte ha något överflödigt parti, vilket innebär att om man kan ta bort ett parti från regeringen och de fortfarande har majoritet, så ska den regeringsformationen inte räknas.

Programmet ska fråga efter antalet partier N , där $2 \leq N \leq 35$, och sedan efter antalet mandat för varje parti (alltid ett heltal). Varje parti har minst ett mandat och det totala antalet mandat överstiger inte 1000.

Programmet ska skriva ut ett tal: antalet möjliga regeringsformationer enligt ovan. Svaret kommer inte att överstiga 2 miljarder.

Körningsexempel 1

Antal partier ? 5
Parti 1 ? 3
Parti 2 ? 5
Parti 3 ? 8
Parti 4 ? 4
Parti 5 ? 2

Svar: 4

Förklaring: Regeringen måste ha minst 12 mandat för majoritet. De fyra möjliga regeringarna har:

- 3 + 5 + 4 mandat
- 5 + 8 mandat
- 8 + 4 mandat
- 3 + 8 + 2 mandat

Körningsexempel 2

Antal partier ? 10
Parti 1 ? 12
Parti 2 ? 66
Parti 3 ? 39
Parti 4 ? 37
Parti 5 ? 21
Parti 6 ? 31
Parti 7 ? 20
Parti 8 ? 53
Parti 9 ? 20
Parti 10 ? 6

Svar: 71