

Programmeringsolympiaden 2019

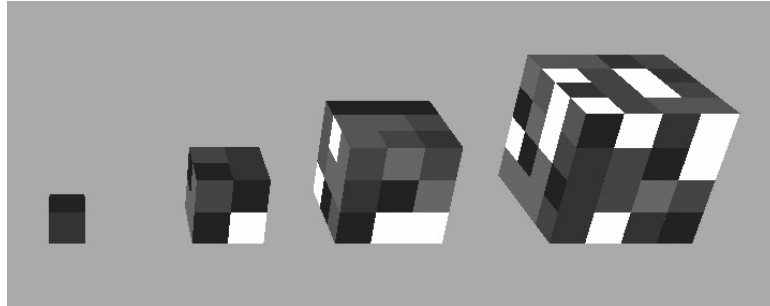
TÄVLINGSREGLER FÖR SKOLKVALET

- Tävlingen äger rum på av skolan bestämt datum under **fyra timmar**. **Ingen förlängning ges för lunch eller raster**. 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 avtalad lokal på skolan.
- Tävlingen består av sex uppgifter som vardera ska lösas genom ett datorprogram i valfritt programmeringsspråk.
- **Indata kan läsas in i programmet på valfritt sätt**, t.ex. genom att programmet för en dialog med användaren (som i körningsexemplena i uppgifterna), att de skrivs in i ett grafiskt gränssnitt eller att datafiler skickas till *standard input*. Kom bara överens med din lärare om hur programmet ska testas.
- Dina lösningar kommer att testköras med förpreparerade indata. Varje uppgift testas normalt med 5 testfall, som vardera ger 1 poäng om ditt program skriver ut korrekt svar inom en exekveringstidsgräns av **3 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 3 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...uppg6 med passande filtillägg. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

Lycka till!

UPPGIFT 1 – KUBER

Nadja klistrar ihop små kuber med sidlängd 1 till större kompakta kuber. Hur många småkuber går det åt om hon vill ha en kub av varje sort med sidlängd från 1 till N ? Figuren nedan visar de fyra kuberna i det första exemplet ($N = 4$). Tillsammans innehåller de $1 + 8 + 27 + 64 = 100$ småkuber.



Programmet ska fråga efter talet N , där $2 \leq N \leq 40$, och skriva ut antalet småkuber som går åt.

Körningsexempel 1

Talet N ? 4
Antal: 100

Körningsexempel 2

Talet N ? 7
Antal: 784

UPPGIFT 2 – STUDSCHIFFER

Fretchif har hittat på ett revolutionärt chiffer som ingen kommer kunna lösa! Så här går det till: Man väljer två heltal N och M och ritar upp ett rutnät med N rader och M kolumner. Därefter börjar man skriva in strängen man krypterar, en bokstav i taget, snett nedåt från det översta vänstra hörnet (se figuren). När man kommer till en av rutnätets väggar, så låter man "bokstavsstrålen" studsas mot väggen (se figuren). Ifall man vid något tillfälle hamnar på en ruta som det redan står en bokstav i, så sparar man den bokstav man tänkte skriva, och skriver den i nästa lediga ruta man kommer till istället. När alla bokstäver i strängen som krypteras är slut så läser man av rutnätet rad för rad, och det blir det krypterade meddelandet.

Givet ett krypterat meddelande och storleken på rutnätet som användes vid krypteringen, skriv ut originalmeddelandet.

Notera att vissa meddelanden inte går att kryptera med vissa storlekar på rutnät, då det är möjligt att man aldrig kommer till någon ny ledig ruta men fortfarande har bokstäver kvar att placera ut. I givna testfall är det dock garanterat att detta inte händer.

Ett exempel: Säg att strängen vi ska kryptera är ABCDEFGHIJKLMNOPQRST, och att rutnätet har storlek 6×13 . Figuren visar hur rutnätet ser ut vid olika tidpunkter under krypteringen. Det krypterade meddelandet blir ATKBSJLCRIMDHNEGQOFP.

A												
	B											
		C										
			D									
				E								
					F							

Efter 6 bokstäver är skrivna

A												
	B											
		C										
			D				H					
				E		G						
					F							

Efter 8 bokstäver är skrivna, vi har nu studsat en gång i bottenväggen

A										K		
	B								J		L	
		C						I				M
			D				H				N	
				E		G						
					F							

Efter 14 bokstäver är skrivna, vi har nu även studsat i övre och högra väggen

A				T						K		
	B				S				J		L	
		C				R		I				M
			D				H				N	
				E		G		Q		O		
					F				P			

Efter alla 20 bokstäver är skrivna, notera att vi inte skrev över H:et med ett R, utan vi skrev R:et på nästa lediga plats

Programmet ska fråga efter antalet rader N och kolumner M i rutnätet ($2 \leq N \leq 20$ och $2 \leq M \leq 20$). Därefter ska programmet fråga efter det krypterade meddelandet, en sträng med högst 30 bokstäver, valda bland A till Z (endast stora bokstäver). Det är garanterat att det finns en möjlig ursprungssträng som ger denna chiffrerad text. Programmet ska skriva ut denna ursprungssträng.

Poängsättning:

För testfall värda 1 poäng gäller att $N = 2$ och att bokstavsstrålen aldrig når högra väggen

För testfall värda ytterligare 2 poäng gäller att bokstavsstrålen aldrig når högra väggen

Körningsexempel 1

Antal rader ? 2

Antal kolumner ? 20

Krypterad text ? PORMEIGRGAMRN

Meddelande: PROGRAMMERING

Körningsexempel 2 (se figuren ovan)

Antal rader ? 6

Antal kolumner ? 13

Krypterad text ? ATKBSJLCRIMDHNEGQOFP

Meddelande: ABCDEFGHIJKLMNOPQRST

Körningsexempel 3

Antal rader ? 5

Antal kolumner ? 7

Krypterad text ? SLUIMPEGEHTR

Meddelande: SUPERHEMLIGT

Körningsexempel 4

Antal rader ? 15

Antal kolumner ? 19

Krypterad text ? DALIGREKTANGEL

Meddelande: DALIGREKTANGEL

UPPGIFT 3 – RENOVERINGEN

Johanna håller på att renovera hemma i sin lägenhet. Hon lämnar inget åt slumpen utan har planerat exakt hur många spikar hon behöver under renoveringen.

Totalt sett behöver hon N spikar med längderna x_1, x_2, \dots, x_N .

I sin spiklåda har hon M spikar med längderna y_1, y_2, \dots, y_M .

Om Johanna behöver en spik med längd x_i kan hon använda en spik med längd y_j om $x_i \leq y_j$ eftersom hon kan kapa av den längre spiken tills den är precis så lång som behövs. Däremot kan hon inte kombinera två korta spikar till en längre spik, eller kapa en spik flera gånger (den har ju bara ett spikhuvud).

Skriv ett program som beräknar hur många spikar Johanna behöver köpa, och vilka längder dessa spikar ska ha. Hon vill köpa så få spikar som möjligt. Om det finns flera sätt att köpa minimalt antal spikar, så vill hon köpa spikar av så kort total längd som möjligt.

Programmet ska fråga efter N och M , antalet spikar Johanna behöver och antalet spikar hon har ($1 \leq N \leq 15$ och $1 \leq M \leq 15$). Därefter ska programmet läsa in de N längderna på de spikar Johanna behöver, och slutligen de M längderna på de spikar hon har. Samtliga längder är heltal mellan 1 och 100.

Programmet ska först skriva ut det minsta antalet spikar Johanna behöver köpa. På nästa rad ska programmet skriva ut längderna på spikarna Johanna ska köpa, i stigande ordning.

Poängsättning: För testfall värda 2 poäng gäller att Johanna behöver använda alla spikar hon redan har, och hon behöver inte kapa någon av dessa.

Körningsexempel 1

```
Behöver antal? 6
Har antal? 3
Behöver längd ? 64
Behöver längd ? 13
Behöver längd ? 45
Behöver längd ? 28
Behöver längd ? 82
Behöver längd ? 77
Har längd ? 45
Har längd ? 82
Har längd ? 64
```

```
Antal: 3
Längder: 13 28 77
```

Förklaring: Johanna behöver bara fylla på med tre extra spikar av längderna 13, 28 och 77.

Körningsexempel 2

```
Behöver antal? 3
Har antal? 2
Behöver längd ? 11
Behöver längd ? 50
Behöver längd ? 45
Har längd ? 45
Har längd ? 100
```

```
Antal: 1
Längder: 11
```

Förklaring: Johanna behöver köpa en spik av längd 11, och dessutom kapa en spik av längd 100 till 50. Hon hade kunnat köpa en spik av längd 50 och kapa spiken av längd 100 till längd 11, men då hade hon behövt köpa spikar av längre total längd.

UPPGIFT 4 – MULTATIONER

På den ännu upptäckta exoplaneten PO-2019 består invånarnas arvs massa av en sträng, där varje bokstav är antingen A, B eller C. Livets utveckling har gått lite snabbare där än på jorden (exempelvis kan alla lösa programmeringsproblem redan som nyfödda). Anledningen tros vara att istället för vanliga mutationer sker "multationer", som ändrar *alla* förekomster av en viss bokstav samtidigt. Bokstaven byts ut mot en sträng som kan innehålla 1, 2 eller 3 bokstäver (se figuren nedan). Detta gör att längden på arvs massan kan öka ganska snabbt.

Skriv ett program som läser in två strängar S och T och skriver ut den kortaste sekvensen av multationer som ändrar S till T . Ingen av strängarna innehåller mer än 10 bokstäver och varje bokstav är antingen A, B eller C.

Programmet ska skriva ut en rad för varje multation, i den ordningen de sker. Varje rad ska innehålla bokstaven som ändras och strängen som den ändras till. Det kommer alltid att finnas en lösning med högst 3 multationer. Om det finns flera optimala sekvenser kan du ange vilken som helst av dem.

Poängsättning: För testfall värda 1 poäng gäller att S och T är lika långa. I testfall värda 2 poäng finns en optimal sekvens där alla multationer är samma.

Körningsexempel 1

Start ? ABA
Mål ? CBC

Multation 1: A → C

Körningsexempel 2

Start ? BC
Mål ? CACCAB

Multation 1: B → A
Multation 2: C → CAB
Multation 3: A → CA

Körningsexempel 3

Start ? CAC
Mål ? CABCACAB

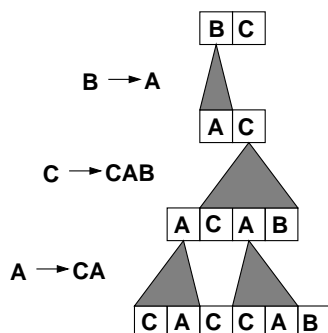
Multation 1: C → AB
Multation 2: A → CA

Körningsexempel 4

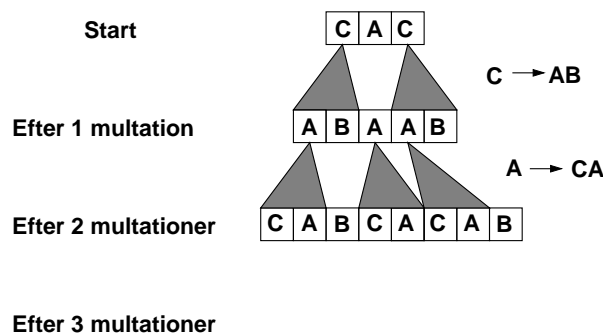
Start ? AABAC
Mål ? AABBBBAC

Multation 1: B → BB
Multation 2: B → BB

Exempel 2



Exempel 3



UPPGIFT 5 – ROBOTTÄVLING

Du har byggt en robot till en robottävling. En av uppgifterna som roboten ska lösa går ut på att räkna antalet kuber i ett kvadratisk rutnät. Rutnätet har N rader och N kolumner, och i varje ruta finns en stapel med 1 till 5 identiska kuber. Tyvärr är din robot inte särskilt bra på att samla information. Det enda den kan göra är att hitta höjden på den högsta stapeln i varje rad och kolumn. Du tänker kompensera för detta genom att göra roboten väldigt smart. Skriv ett program som, givet informationen roboten samlade in, räknar ut det minsta möjliga och det största möjliga totala antalet kuber i rutnätet.

Programmet ska fråga efter rutnätets storlek N . Därefter ska det läsa in N heltal r_i ($1 \leq r_i \leq 5$), höjden av högsta stapeln i rad i , och därefter ytterligare N heltal c_i ($1 \leq c_i \leq 5$), höjden av högsta stapeln i kolumn i . Det är garanterat att det inte finns några motsägelser i indatan, det vill säga det kommer alltid att finnas minst en giltig utplacering av kuber som ger de givna värdena.

Poängsättning

För testfall värda 1 poäng är $N = 3$ och det räcker att få rätt på största antalet kuber.

För testfall värda 1 poäng är $5 \leq N \leq 10$, det räcker att få rätt på största antalet kuber.

För testfall värda 1 poäng är $N = 3$.

För testfall värda 2 poäng är $5 \leq N \leq 10$.

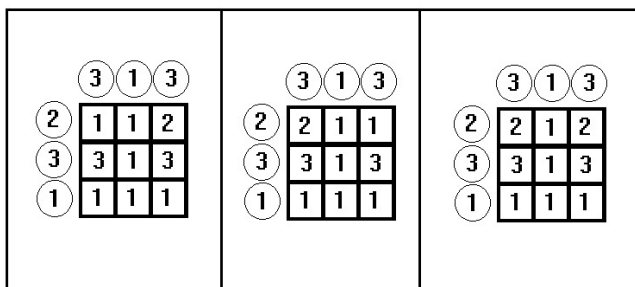
Körningsexempel 1

Storlek ? 3
 Rad 1, högsta stapeln ? 2
 Rad 2, högsta stapeln ? 3
 Rad 3, högsta stapeln ? 1
 Kolumn 1, högsta stapeln ? 3
 Kolumn 2, högsta stapeln ? 1
 Kolumn 3, högsta stapeln ? 3

Min: 14

Max: 15

Förklaring: Figuren nedan visar de enda möjliga konfigurationerna. Båda de två första har totalt 14 kuber och den tredje har totalt 15 kuber.



Körningsexempel 2

Storlek ? 4
 Rad 1, högsta stapeln ? 5
 Rad 2, högsta stapeln ? 5
 Rad 3, högsta stapeln ? 5
 Rad 4, högsta stapeln ? 5
 Kolumn 1, högsta stapeln ? 5
 Kolumn 2, högsta stapeln ? 5
 Kolumn 3, högsta stapeln ? 5
 Kolumn 4, högsta stapeln ? 5

Min: 32

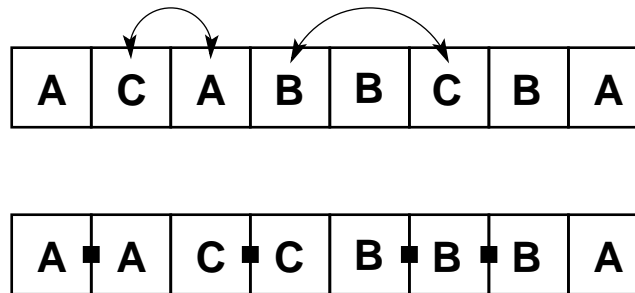
Max: 80

UPPGIFT 6 – AVSLUTNINGSCEREMONIN

Några olika sällskap är inbjudna att sitta på första raden under avslutningsceremonin för årets upplaga av IOI (International Olympiad in Informatics). Varje person har blivit tilldelad en plats. Då organisatörerna av IOI inte insåg att sällskapen gärna sitter tillsammans har de delat ut platserna lite huller om buller. Personerna tänker därför ta situationen i egna händer: Genom att byta platser med varandra försöker de maximera antalet par av personer från samma sällskap som sitter bredvid varandra. Organisatörerna blir arga ifall platsbyttandet blir för stökigt, och bestämmer därför att varje person får byta plats högst en gång, och då med en person som sitter högst K platser bort.

Skriv ett program som beräknar det största antalet par av personer från samma sällskap sittande bredvid varandra som går att uppnå med giltiga platsbyten.

Programmet ska läsa in en sträng, med längd $1 \leq N \leq 30$, som beskriver den ursprungliga raden. Varje bokstav i strängen beskriver vilket sällskap personen på motsvarande plats tillhör, och är antingen A, B, C eller D. Därefter ska programmet fråga efter maxavståndet K som personerna får flytta ($1 \leq K \leq 2$).



FIGUR 1. Lösningen till det andra körningsexemplet. Pilarna visar hur personerna ska byta plats. De svarta kvadraterna markerar par av personer ur samma sällskap som sitter bredvid varandra efter flyttningen.

Poängsättning:

För testfall värda 1 poäng gäller att $K = 1$ och $N \leq 15$ och det finns 3 sällskap (A,B,C).

För testfall värda 2 poäng gäller att $K = 2$ och det finns 2 sällskap (A,B).

För testfall värda 2 poäng gäller att $K = 2$ och det finns 4 sällskap (A,B,C,D).

Körningsexempel 1

Rad ? BAABAA
Maxavstånd ? 1

Max antal par: 3

Förklaring: Placeringen ABBAAA kan uppnås genom att byta plats på den första och den andra personen, och byta plats på den tredje och den fjärde personen.

Körningsexempel 2

Rad ? ACABBCBA
Maxavstånd ? 2

Max antal par: 4

Förklaring: Se figuren ovan.