

Programmeringsolympiaden 2018

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 fem 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...uppg5 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 – TUNNELBANAN

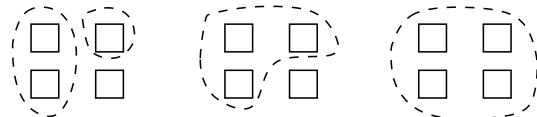
Baltic Olympiad in Informatics – programmeringstävlingen för länderna i östersjöregionen – kommer till våren att anordnas i Stockholm, och deltagarna kommer åka tunnelbana mellan vandrarhemmet och tävlingsarenan.

På tunnelbanetågen finns det sätesgrupper med fyra säten vardera. Nu kommer ett antal grupper av deltagare och vill sätta sig. Varje grupp har storlek 1, 2, 3 eller 4. Helst skulle alla personerna i en grupp vilja sitta i samma firsätesgrupp, alltså slippa dela på sig. Hur många firsätesgrupper krävs för att detta ska vara möjligt?

Skriv ett program som läser in fyra heltal a_1, a_2, a_3, a_4 – antalet grupper av varje storlek (alla tal är mellan 0 och 100) – och skriver ut det minsta antalet firsätesgrupper som behövs.

Körningsexempel 1

Antal 1-grupper ? 1
Antal 2-grupper ? 1
Antal 3-grupper ? 1
Antal 4-grupper ? 1

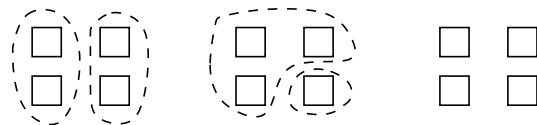


En optimal placering i exempel 1 (streckade linjer avgränsar grupper av personer)

Behövs: 3

Körningsexempel 2

Antal 1-grupper ? 1
Antal 2-grupper ? 2
Antal 3-grupper ? 1
Antal 4-grupper ? 0

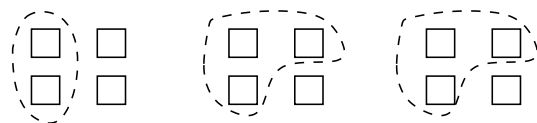


En optimal placering i exempel 2.

Behövs: 2

Körningsexempel 3

Antal 1-grupper ? 0
Antal 2-grupper ? 1
Antal 3-grupper ? 2
Antal 4-grupper ? 0



En optimal placering i exempel 3.

Behövs: 3

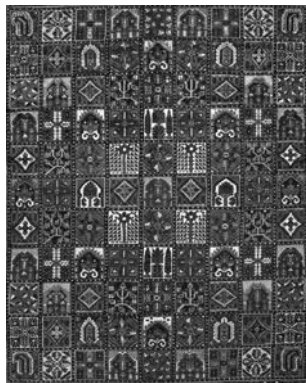
Poängsättning

För testfall värda 1 poäng finns inga grupper av storlek 1.

För testfall värda 1 poäng finns inga grupper av storlek 2.

För testfall värda 1 poäng finns inga grupper av storlek 3.

För testfall värda 2 poäng gäller inga speciella begränsningar.



UPPGIFT 2 – KÖPA MATTA

Under IOI i Teheran i somras märktes en ökad efterfrågan på persiska mattor, i synnerhet sådana vars mönster utgörs av ett rutmönster med $L \times B$ rutor (se bild ovan), eftersom dessa är mycket lämpliga för att i hemlighet prova ut algoritmer på.

En mattas pris brukar avgöras av antalet rutor, så en typisk kund vill ha en matta med minst M rutor och högst N rutor. Om det finns flera möjliga mattor vill kunden ha en så kvadratisk matta som möjligt, d.v.s. den vill att $|L - B|$ är så litet som möjligt.

Skriv ett program som läser in talen M och N och skriver ut det bästa valet av B och L (den minsta sidlängden först). Det kommer alltid finnas ett unikt bästaval.

Körningsexempel 1

Minimalt antal rutor ? 13

Maximalt antal rutor ? 13

Bästa mått: 1 13

Körningsexempel 2

Minimalt antal rutor ? 60

Maximalt antal rutor ? 70

Bästa mått: 8 8

Körningsexempel 3

Minimalt antal rutor ? 750

Maximalt antal rutor ? 755

Bästa mått: 26 29

Poängsättning

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

För testfall värda 1 poäng gäller att $M = N$, och $10^{11} \leq M \leq 10^{12}$

För testfall värda 2 poäng gäller att $10^8 \leq M \leq N \leq 10^{12}$

UPPGIFT 3 – FLYTTKARTONGER

Du har just hjälpt en kompis att flytta, men tyvärr har du fastnat i fel ände av en smal korridor full med flyttkartonger. Korridoren består av N staplar av flyttkartonger, där stapel nummer i innehåller a_i kartonger. Alla kartonger är lika stora.

Det enda sättet att ta sig ut är att gå ovanpå staplarna från stapel 1 till stapel N . Om man befinner sig på en stapel kan man gå till en närliggande stapel, men bara om den inte är högre än den man står på. Om stapeln man står är minst två kartonger högre än en närliggande stapel kan man dessutom knuffa ner den översta kartongen från stapeln man står på till den närliggande stapeln. Detta kan upprepas hur många gånger som helst.

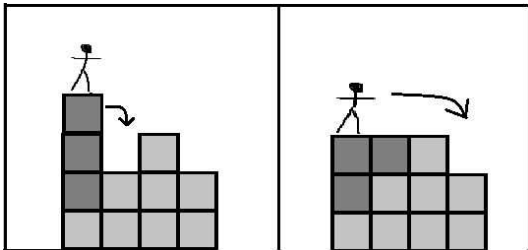
Du befinner dig just nu på stapel 1. Tyvärr kanske det är omöjligt för dig att komma till stapel N . Men som tur är får du lägga till valfritt antal *extra* kartonger till stapel 1 innan du börjar gå. Skriv ett program som beräknar hur många extra kartonger du behöver lägga till för att kunna ta dig till stapel N .

Programmet ska läsa in antalet staplar N , och sedan höjden a_i på varje stapel. Programmet ska skriva ut det minsta antalet extra kartonger som behöver läggas till.

Körningsexempel 1

Antal staplar ? 4
 Stapel 1 ? 1
 Stapel 2 ? 2
 Stapel 3 ? 3
 Stapel 4 ? 2

Extra kartonger: 3



Bilden visar exempel 1. De mörkgrå kartongerna är extrakartonger. Strategin är att knuffa ner den översta extrakartongen till stapel 2. Därefter kan man gå raka vägen till stapel 4. Det inte hade fungerat med färre än 3 extrakartonger.

Körningsexempel 2

Antal staplar ? 3
 Stapel 1 ? 5
 Stapel 2 ? 2
 Stapel 3 ? 3

Extra kartonger: 0

Körningsexempel 3

Antal staplar ? 6
 Stapel 1 ? 30
 Stapel 2 ? 5
 Stapel 3 ? 10
 Stapel 4 ? 15
 Stapel 5 ? 13
 Stapel 6 ? 30

Extra kartonger: 261

Poängsättning

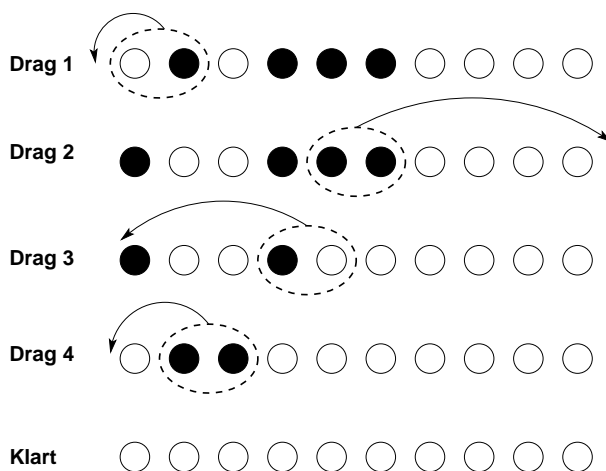
För testfall värda	1 poäng	gäller att	$N = 3$, $1 \leq a_i \leq 20$.
	1 poäng		$3 \leq N \leq 5$, $1 \leq a_i \leq 100$.
	1 poäng		$5 \leq N \leq 10$, $1 \leq a_i \leq 1000$.
	2 poäng		$10 \leq N \leq 20$, $1 \leq a_i \leq 3000$.

UPPGIFT 4 – BRICKOR

Karin har hittat på ett ensamspel som spelas med Othello-brickor, vilka är svarta på ena sidan och vita på andra sidan. Hon lägger ut en rad med brickor, som var och en kan vara svart eller vit. Målet är att få alla brickor att ha den vita sidan uppåt.

Ett “drag” är att “plocka ut” ett intilliggande par av brickor någonstans ur sekvensen, vända på dem (vit blir svart, svart blir vit), och lägga tillbaka dem antingen i början av raden eller i slutet av raden, utan att ändra parets inbördes ordning.

Skriv ett program som, givet den ursprungliga raden av brickor (en sträng med enbart bokstäverna S och V och längd mellan 3 och 15 tecken), skriver ut det minsta antalet drag som behövs för att göra alla brickor vita. För givna testdata kommer det alltid vara möjligt att nå målet.



En möjlig dragsekvens i exempel 2

Körningsexempel 1

Brickor ? SVVSVVV

Minst antal drag: 2

Körningsexempel 2

Brickor ? VSVSSSVVVV

Minst antal drag: 4

Poängsättning

För testfall värda 3 poäng är strängens längd högst 10, och svaret högst 6.
För testfall värda 2 poäng gäller inga speciella begränsningar.

UPPGIFT 5 – DUTUB

Du vet att du borde lägga dig vid det här laget! Men du ska bara kolla *lite* mer på DuTub innan du lägger dig.

Det finns ett antal *kategorier* du är intresserad av. Varje video på DuTub kan tillhöra en eller flera kategorier. Innan du lägger dig måste du ha sett minst en video i varje kategori. Men du vill förstås inte vara uppe längre än nödvändigt.

Skriv ett program som, givet en lista på videor, beräknar den minsta möjliga tid du måste titta för att ha sett minst en video från varje kategori.

Programmet ska läsa in antalet videor N och sedan, för var och en av de N videorna, dess längd i sekunder (ett heltal mellan 1 och 900) samt en sträng som anger de kategorier som videon tillhör. Varje bokstav (som är mellan a och j) betecknar en kategori. Varje video tillhör minst en kategori, och inga kategorier upprepas i en videos beskrivning. Det kan sammanlagt finnas högst 10 olika kategorier. Programmet ska skriva ut det minsta antalet sekunder du behöver spendera på DuTub innan du sett videor ur alla kategorier.

Körningsexempel 1

```

Antal videor ? 5
Video 1, längd      ? 200
Video 1, kategorier ? ei
Video 2, längd      ? 150
Video 2, kategorier ? e
Video 3, längd      ? 10
Video 3, kategorier ? g
Video 4, längd      ? 50
Video 4, kategorier ? gb
Video 5, längd      ? 60
Video 5, kategorier ? i
    
```

Svar: 250

Körningsexempel 2

```

Antal videor ? 6
Video 1, längd      ? 268
Video 1, kategorier ? abe
Video 2, längd      ? 271
Video 2, kategorier ? ca
Video 3, längd      ? 262
Video 3, kategorier ? da
Video 4, längd      ? 145
Video 4, kategorier ? cd
Video 5, längd      ? 150
Video 5, kategorier ? ebc
Video 6, längd      ? 143
Video 6, kategorier ? deb
    
```

Svar: 412

Förklaring av körningsexempel 1: Här finns det totalt 4 kategorier: e, i, g, b, som står för ekorrar, igelkottar, getter och bumbibjörnar. Den tredje videon (g) ger ingenting, för vi måste ändå se den fjärde videon (gb), då den är den enda som innehåller bumbibjörnar. Vi föredrar video 1 framför video 2+5, eftersom dess totala tid är kortare och täcker in båda kategorierna igelkottar och ekorrar. Svaret är alltså 250 sekunder, genom att titta på den första och fjärde videon.

Poängsättning

För testfall värda 1 poäng gäller $1 \leq N \leq 10$, och varje video tillhör bara en kategori.

För testfall värda 2 poäng gäller $1 \leq N \leq 10$.

För testfall värda 2 poäng gäller $N = 30$.