

Programmeringsolympiaden 2026

TÄVLINGSREGLER FÖR SKÖLKVALET

- Tävlningen 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.
- Tävlningen 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**. Ingen test av indata behöver göras, den följer specifikationerna i uppgiften.
- 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.
- 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. **Det är heller inte tillåtet att använda generativ AI såsom ChatGPT eller Github Copilot.** Miniräknare är tillåten.
- Tävlingsbidraget ska lämnas in i form av källkodsfiler (uppg1...uppg6 med passande filtillägg) som läggs på överenskommen plats. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

De högst placerade i kvalet går vidare till finalen i Göteborg där landslagsplatser till BOI i Finland och IOI i Uzbekistan står på spel. Lycka till!

UPPGIFT 1 – FRACK JACK

Alice och Bertil kastar boll i en park. Exakt 21 meter framför Alice och Bertil finns en stolpe och Jack ger en utmaning till Alice och Bertil.

Utmaningen är att Alice och Bertil ska kasta varsin boll så nära stolpen som möjligt, men bollen får inte kastas förbi stolpen. Den som kastar närmast 21 meter, utan att kasta förbi 21 meter, vinner. Om både Alice och Bertil kastar över 21 meter, eller om de kastar lika långt så vinner Jack istället. De har nu kastat och behöver hjälp att avgöra vem som vann.

Skriv ett program som först läser in talen A och B , antalet meter som Alice och Bertil har kastat vardera.

Det är garanterat att A och B uppfyller $0 \leq A \leq 30$ och $0 \leq B \leq 30$.

Sedan ska programmet skriva ut vem som vann: antingen "Alice", "Bertil" eller "Jack". Notera att du inte ska skriva ut citationstecken.

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar:

- (1) $A \neq B$ och $A, B \leq 21$
- (2) $A, B \leq 21$
- (3) Jack vann inte i det här testfallet.
- (4-5) Inga ytterligare begränsningar.

Körningsexempel 1

Alice kast ? 7
Bertils kast ? 5

Svar: Alice

Förklaring: Alice är närmare 21, därmed vann hon.

Körningsexempel 2

Alice kast ? 26
Bertils kast ? 22

Svar: Jack

Förklaring: Eftersom båda kastade längre än 21 meter vann Jack.

UPPGIFT 2 – SNURRA KOMPASS

Du har en kompass som kan peka i 360 olika riktningar. Just nu pekar den rakt uppåt, på talet 0.

Tyvärr så fungerar inte kompassen! Det enda sättet du kan få den att peka i andra riktningar är genom att lyfta på hela kompassen och rotera den ett antal grader motsols. Till din hjälp har du två robotarmar. Dessa är tyvärr inte perfekta: den första robotarmen kan snurra kompassen exakt a grader motsols och den andra robotarmen kan snurra b grader motsols. Robotarmarna kan endast rotera kompassen exakt a respektive b grader varje gång de används, varken mer eller mindre. På grund av detta går det kanske inte att få kompassen att peka i alla 360 riktningar.

Detta får dig att undra hur många unika riktningar kompassen kan peka åt? Kompassen kan endast roteras genom att använda robotarmarna ett flertal gånger.

Skriv ett program som läser in talen a och b , antalet grader som den första respektive andra robotarmen snurrar kompassen.

Det är garanterat att a och b uppfyller $1 \leq a \leq 360$ och $1 \leq b \leq 360$.

Sedan ska programmet skriva ut antalet unika riktningar som kompassen kan peka åt, om man snurrar den ett flertal gånger med robotarmarna.

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar:

- (1) $a = 13$
- (2-3) $a = b$
- (4-5) Inga ytterligare begränsningar.

Körningsexempel 1

Gradtal a ? 50
Gradtal b ? 180

Svar: 36

Förklaring: Med denna uppsättningen robotarmar går det att få kompassen att peka på varje heltalsmultipel av 10: 0, 10, 20, ..., 350. Om du tar robotarmen med $a = 50$ och använder den 8 gånger kan du exempelvis få kompassen att peka 40 grader motsols.

Körningsexempel 2

Gradtal a ? 360
Gradtal b ? 360

Svar: 1

Förklaring: Hur många gånger du än roterar kompassen med denna uppsättning robotarmar kommer den fortsätta vara riktad 0 grader motsols.

UPPGIFT 3 – ARBORIST

Teo har äntligen landat sitt drömjobb; han har börjat jobba som arborist! En viktig del av en arborists arbete är att plantera träd.

Teo behöver nu hjälp med att plantera N träd på heltalspositioner längs en tallinje. Tallinjen börjar vid position 0 och värdet ökar med 1 för varje steg åt höger (och minskar med 1 åt vänster).

För alla N träd har Teo redan bestämt exakt vilka heltalspositioner på tallinjen som de ska planteras på. Dessa positioner är alla positiva heltal.

Teo befinner sig just nu vid position 0, tillsammans med N stora trädplantor. För att plantera träden kan han göra följande:

- Förflytta sig ett steg åt höger (öka positionen med 1).
- Förflytta sig ett steg åt vänster (minska positionen med 1).
- Om Teo befinner sig vid position 0 kan han plocka upp en av de N trädplantorna.
- Ta en av trädplantorna han bär på och plantera den vid positionen han befinner sig.

Varje sådan handling tar exakt en sekund.

Tyvärr klarar Teo av att bära som mest K kilogram träd åt gången. Varje träd väger antingen 20 eller 40 kilogram. När alla träd har planterats måste Teo dessutom återvända till position 0.

Din uppgift är att beräkna hur lång tid det tar för Teo att plantera alla träd och sedan återvända till position 0, om han arbetar på snabbast möjliga sätt.

Skriv ett program som först läser in N och K ($1 \leq N \leq 8$, $20 \leq K \leq 50$), antalet träd som ska planteras, respektive antalet kilogram som Teo orkar bära. Därefter ska programmet läsa in på nästa rad N heltal $x_1, \dots, x_N$ ($1 \leq x_i \leq 30$), positionerna där varje trädplanta ska planteras. På sista raden ska det läsas in N heltal $k_1, \dots, k_N$ ($k_i \leq K$), vikten av varje trädplanta i kilogram.

Inmatningen beskriver alltså N träd numrerade 1 till N , där träd i ska planteras på position x_i och väger k_i kilogram. Varje k_i är antingen 20 eller 40.

Programmet ska sedan skriva ut antalet sekunder det tar för Teo att plantera alla träd och därefter återvända till 0, givet att han gör detta på det snabbast möjliga sätt.

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar:

- (1) $K = 25$
- (2) $k_i = 20$ för alla i .
- (3) Alla träd ska planteras på samma position.
- (4-5) Inga ytterligare begränsningar.

Körningsexempel finns på nästa sida.

Körningsexempel 1

N och K ? 2 45

Trädens positioner ? 1 2

Trädens vikt ? 20 40

Svar: 10

Förklaring: Det finns två träd i detta körningsexempel. Ett av träden väger 20 kilogram och ska planteras vid position 1. Det andra trädet väger 40 kilogram och ska planteras vid position 2. I körningsexemplet är en optimal lösning för Teo att göra följande:

- Plocka upp trädet som ska planteras vid position 1.
- Gå höger.
- Plantera trädet som ska planteras vid position 1.
- Gå vänster.
- Plocka upp trädet som ska planteras vid position 2.
- Gå höger.
- Gå höger.
- Plantera trädet som ska planteras vid position 2.
- Gå vänster.
- Gå vänster. Han har nu återvänt till position 0 och är därmed färdig.

Det finns andra sätt att plantera träden lika snabbt, men det går inte att göra det på mindre än 10 sekunder.

Körningsexempel 2

N och K ? 4 42

Trädens positioner ? 10 9 7 8

Trädens vikt ? 20 40 20 20

Svar: 60

Förklaring: I det här körningsexemplet är en optimal lösning att först plantera trädet vid position 9, sen träden vid position 10 och 8 i en och samma tur, till sist det vid position 7.

UPPGIFT 4 – TINGSPLATSEN

Arkeologen Arvid är på jakt efter en okänd tingsplats från järnåldern någonstans i Mellansverige. Han har kartlagt ett antal gårdar som fanns i området och har kommit fram till att tingsplatsen måste ligga på samma avstånd till de olika gårdarna.

Området som Arvid undersökt kan representeras av ett rutnät med N rader och M kolumner. Det tar en minut för en järnåldersmänniska att ta sig mellan två närliggande rutor. Två rutor är närliggande om de delar en av sina fyra sidor.

Du får givet en karta av rutnätet som markerar alla positioner av gårdar. Det är känt att det tar lika lång tid att ta sig från var och en av gårdarna till den okända tingsplatsen.

Din uppgift är att skriva ett program som hittar vilka rutor där tingsplatsen skulle kunna funnits.

Programmet ska först läsa in heltalen N och M ($1 \leq N, M \leq 10$), antalet rader och kolumner i rutnätet. Därefter ska programmet läsa in N rader med M tecken vardera, som innehåller rutnätets rader. Tecknet "*" betyder att det fanns en gård på motsvarande plats i rutnätet, "." betyder att det inte fanns en gård.

Det är garanterat att det finns minst två gårdar i rutnätet, och det finns alltid minst en ruta där tingsplatsen kan vara

Programmet ska sedan skriva ut en kopia av rutnätet från indatan, men de rutor som tingsplatsen kan vara belägen på är markerat med "X".

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar utöver de som redan nämnts:

- (1) $N = 1$
- (2) Det finns exakt 4 gårdar som alla ligger på hörnen av samma rektangel.
- (3-5) Inga ytterligare begränsningar.

Körningsexempel finns på nästa sida.

Körningsexempel 1

```
N M ? 4 5
Rad 1 ? *...*
Rad 2 ? .....
Rad 3 ? .....
Rad 4 ? .....
```

```
Svar:
*.X.*
..X..
..X..
..X..
```

Förklaring av körningsexempel 1: Gårdarna ligger i första och sista kolumnen på översta raden. Den tredje rutan på översta raden ligger två minuter från båda gårdarna (två steg åt vänster respektive två steg åt höger). Alla rutor rakt under den har samma avstånd till båda gårdarna: gå upp till översta raden och sedan lika åt vänster eller höger. Därför markeras hela den kolumnen med "x". Övriga rutor har olika avstånd och markeras inte.

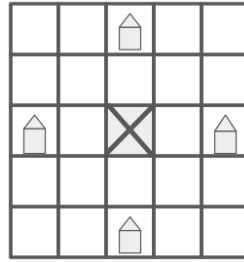
Körningsexempel 2

```
N M ? 1 10
Rad 1 ? *.....*
```

```
Svar:
*...X*.....
```

Förklaring: Gårdarna ligger i första och sjunde kolumnen. Rutan i fjärde kolumnen ligger tre minuter ifrån båda gårdarna. Ingen annan ruta på raden har samma avstånd till båda, därför markeras endast en ruta med "x".

Körningsexemplen fortsätter på nästa sida.



Bilden visar lösningen till exempel 3. Den markerade rutan i mitten är den enda möjliga positionen hos tingsplatsen. Det tar två minuter att ta sig från varje gård till den här rutan.

Körningsexempel 3

```
N M ? 5 5
Rad 1 ? ..*..
Rad 2 ? .....
Rad 3 ? *...*
Rad 4 ? .....
Rad 5 ? ..*..
```

Svar:

```
..*..
.....
*.X.*
.....
..*..
```

Förklaring: Den markerade rutan i mitten är den enda möjliga positionen för tingsplatsen. Det tar två minuter att ta sig till varje gård från den här rutan.

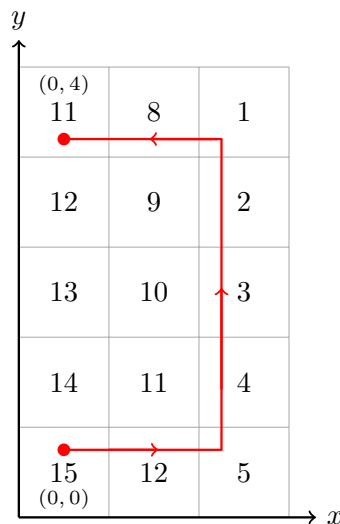
UPPGIFT 5 – LÅNGA TRAPPOR

Det sägs att uppe i Alperna bor C sysstrar som vardera har ett eget favorittal f . Varje syster har därför byggt ett hus f höjdenheter över vattenytan. Husen är byggda på ett $R \times C$ stort rutnät, sådan att rutan längst ner till vänster är $(0, 0)$, och rutan längst upp till höger är $(C - 1, R - 1)$. Den första systemen har byggt sitt hus vid $(0, 0)$, den andra vid $(1, 0)$, och så vidare så att sista systemen har byggt sitt hus vid $(C - 1, 0)$. Med andra ord är husen byggda längs rutnätets bottenrad.

När husen befinner sig så högt uppe blir det svårt att ta sig upp och ner. Därför har varje syster byggt en trappa från sitt hus till de $R - 1$ raderna ovanför, där höjden minskar med 1 för varje trappsteg. För varje ruta som inte är på bottenraden, finns det alltså en ruta nedanför, där höjden är 1 mer än rutan ovan. Till exempel: om systemens hus ligger vid $(2, 0)$ och har höjd 10, byggs ett trappsteg vid $(2, 1)$ med höjd 9, ett steg vid $(2, 2)$ med höjd 8, och så vidare.

Du befinner dig just nu på detta rutnät. Mer exakt befinner du dig på koordinaterna (s_x, s_y) och vill ta dig till (g_x, g_y) . Du kan röra dig till rutor som är intilliggande antingen upp, vänster, ner eller höger. Det är aldrig ett problem att gå höger eller vänster, även om det kan finnas stora höjdskillnader. Självfallet kan du inte röra dig till en koordinat utanför rutnätet.

Höga höjder är ganska jobbiga, så ditt mål är att minimera den tid du spenderar på hög höjd. Mer exakt, om du under din vandring från (s_x, s_y) till (g_x, g_y) besöker en viss mängd rutor, så definierar vi jobbigheten av vandringen som summan av höjden av alla rutor du besöker. Om du vandrar optimalt, vad är minsta möjliga jobbigheten?



Bilden visar hur rutnätet ser ut i körningsexempel 1. Raden längst ner innehåller sysstrarnas hus. Start och mål är även markerade, samt den minst jobbiga vandringen mellan dem.

Problembeskrivningen fortsätter på nästa sida.

Skriv ett program som först läser in heltalen R och C ($1 \leq R \leq 10^9$, $1 \leq C \leq 10$), rutnätets dimensioner. Det ska sedan läsa in C heltal $f_1, \dots, f_N$ ($R - 1 \leq f_i \leq 10^9$), höjden där vardera systems hus befinner sig på. f_1 är höjden på huset vid $(0, 0)$, f_2 vid $(1, 0)$, och så vidare.

Notera att höjderna f_i är valda så att resterande höjder i rutnätet inte är negativa.

Läs sedan in heltalen s_x och s_y ($0 \leq s_x < C$, $0 \leq s_y < R$), koordinaten där du börjar. Läs till sist in heltalen g_x och g_y ($0 \leq g_x < C$, $0 \leq g_y < R$), koordinaten du vill ta dig till.

Programmet ska sedan skriva ut jobbigheten av den minst jobbiga vandringen från start till koordinaterna du vill ta dig till. **Notera: svaret ryms inte nödvändigtvis i ett 32-bitars heltal.**

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar utöver de som redan nämnts:

- (1) $C = 2$
- (2-3) $R \leq 10^4$
- (4-5) Inga ytterligare begränsningar.

Körningsexempel 1

```
R C ? 5 3
Bottenraden ? 15 12 5
Start-x och start-y ? 0 0
Mål-x och mål-y ? 0 4
```

Svar: 61

Förklaring: En bild av hur detta rutnät ser ut finns tidigare i uppgiftsbekrivningen. Den minst jobbiga vandringen går höger, höger, upp, upp, upp, upp, vänster, vänster. Jobbigheten är $15 + 12 + 5 + 4 + 3 + 2 + 1 + 8 + 11 = 61$.

Körningsexemplen fortsätter på nästa sida.

Körningsexempel 2

```
R C ? 5 3
Bottenraden ? 5 20 25
Start-x och start-y ? 1 4
Mål-x och mål-y ? 2 0
```

Svar: 76

Körningsexempel 3

```
R C ? 100000 6
Bottenraden ? 218849 99999 100019 100003 100000 264319
Start-x och start-y ? 1 8493
Mål-x och mål-y ? 5 99998
```

Svar: 4186884119

UPPGIFT 6 – BINÄRA PALINDROM

Mikaela studerar intressanta positiva heltal, och vill nu undersöka tal som är *binära palindrom*. För att formellt beskriva detta definierar vi följande funktioner:

$\text{bin}(n)$ = den binära representationen av heltalet n utan inledande nollor,
 $\text{reverse}(s)$ = strängen s skriven baklänges.

Ett heltal n är alltså ett *binärt palindrom* om

$$\text{bin}(n) = \text{reverse}(\text{bin}(n)).$$

Notera att ett palindrom är en sträng som läses likadant framifrån som bakifrån.

Mikaela är särskilt intresserad av de binära palindrom som innehåller exakt k ettor i dess binära representation, och vill nu veta hur många sådana tal som finns inom intervallet $[l, r]$.

Givet tre heltal l , r och k , bestäm hur många heltal x som uppfyller:

- (1) Talet x är mellan l och r , alltså att $l \leq x \leq r$,
- (2) Talet x är ett tal som Mikaela gillar (dvs. ett binärt palindrom),
- (3) Binärrepresentationen av x innehåller exakt k ettor.

Skriv ett program som läser in heltalen l, r, k ($1 \leq l \leq r \leq 10^{18}, 1 \leq k < 60$).

Programmet ska sedan skriva ut antalet heltal i intervallet $[l, r]$ som är ett binärt palindrom och har exakt k ettor i sin binära representation.

Poängsättning:

De fem testfallen som vardera ger 1 poäng har följande begränsningar utöver de som redan nämnts:

- (1) $l, r \leq 10^9$
- (2) $k \leq 4$
- (3-5) Inga ytterligare begränsningar.

Körningsexempel 1

l r k ? 7 10 2

Svar: 1

Körningsexemplen fortsätter på nästa sida.

Förklaring av körningsexempel 1: Vi söker tal i intervallet $[7, 10]$ som både är ett binärt palindrom och har exakt två ettor i sin binära representation.

Det finns endast 4 tal i intervallet, och deras binära representation är:

$$\begin{aligned} 7 &= 111_2 \quad (\text{palindrom, tre ettor}) \\ 8 &= 1000_2 \quad (\text{ej palindrom}) \\ 9 &= 1001_2 \quad (\text{palindrom, två ettor}) \\ 10 &= 1010_2 \quad (\text{ej palindrom}) \end{aligned}$$

Det enda talet som uppfyller båda villkoren är talet 9, så svaret är 1.

Körningsexempel 2

l r k ? 1 6 2

Svar: 2

Förklaring: Vi söker tal i intervallet $[1, 6]$ som både är ett binärt palindrom och har exakt två ettor i sin binära representation.

Det finns endast 6 tal i intervallet:

$$\begin{aligned} 1 &= 1_2 \quad (\text{palindrom, en etta}) \\ 2 &= 10_2 \quad (\text{ej palindrom}) \\ 3 &= 11_2 \quad (\text{palindrom, två ettor}) \\ 4 &= 100_2 \quad (\text{ej palindrom}) \\ 5 &= 101_2 \quad (\text{palindrom, två ettor}) \\ 6 &= 110_2 \quad (\text{ej palindrom}) \end{aligned}$$

Talen som uppfyller båda villkoren är talen 3 och 5, så svaret är 2.

Körningsexempel 3

l r k ? 999 99999 6

Svar: 94