

Programmeringsolympiaden 2024

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**. 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. Förbudet på generativ AI gäller endast skolkvalet och inte onlinekvalet.** Räknedosa 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.

Lycka till!

UPPGIFT 1 – REDUPLIKATION

Reduplikation är när man upprepar ett ord flera gånger, ofta för att förstärka eller ändra betydelsen. I vissa språk som indonesiska är reduplikation väldigt vanligt, och där har det många olika grammatiska funktioner. Även i svenska används reduplikation ibland, till exempel är det vanligt att man säger "hej hej" istället för "hej".

Rama är en flitig användare av reduplikation när han chattar med sina kompisar. Han skriver ibland samma ord upp till nio gånger, vilket tar lång tid. Därför brukar han istället skriva ordet en gång, åtföljt av en siffra som visar hur många gånger ordet ska upprepas.

Skriv ett program som läser in ett ord som Rama tänker upprepa (som består av som mest 9 bokstäver) och hur många gånger det ska upprepas (som mest 9). Programmet ska sedan skriva ut ordet upprepat rätt antal gånger.

Poängsättning: För testfall värda 1 poäng (utav 5 möjliga) gäller att antalet gånger ordet ska upprepas är 1.

Körningsexempel 1

Ordet ? hej
Antal upprepningar ? 3

Svar: hejhejhej

Körningsexempel 2

Ordet ? ha
Antal upprepningar ? 4

Svar: hahahaha

UPPGIFT 2 – TRE I RAD

Matematikern Lenore Oljer har gjort så mycket matte att hon har tröttnat på primtal. Hon har nu börjat studera så kallade trevliga tripletter. Det är väldigt lätt att skapa en trevlig tripplett. Börja med ett heltal a som är större än noll och skapa sedan trippletten $(a, a + 1, a + 2)$. Alla trippletter som skapats på detta sätt är trevliga. Några exempel på trevliga trippletter är $(4, 5, 6)$ och $(15, 16, 17)$.

För att lära sig mer om trevliga trippletter vill Lenore hitta hur många tal som kan skrivas som produkten av alla talen i en trevlig tripplett. Hon kallar dessa tal "treiga tal". Några exempel på treiga tal är $24 = 2 \cdot 3 \cdot 4$ och $336 = 6 \cdot 7 \cdot 8$. Lenore vill nu veta hur många treiga tal det finns som är mindre än talet N .

Skriv ett program som läser in talet N och skriver ut hur många tal som är mindre än N och är treiga.

Poängsättning:

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

- (1) $N \leq 25$
- (2) $N \leq 80$
- (3) $N \leq 1000$
- (4) $N \leq 10^5$
- (5) $N \leq 10^9$

Körningsexempel 1

$N ? 75$

Svar: 3

Förklaring: De treiga talen mindre än 75 är 6, 24 och 60.

Körningsexempel 2

$N ? 24$

Svar: 1

Förklaring: Eftersom det treiga talet 24 inte är mindre än N räknar vi bara 6.

Körningsexempel 3

$N ? 1234$

Svar: 9

UPPGIFT 3 – MEDDELANDE

I staden som Paula bor i finns ett stort torg. Torget kan representeras av ett rutnät med n rader och m kolumner. I vissa av rutorna finns en bokstav inristad.

Enligt legenden så bildar dessa bokstäver ett meddelande om man går på rutorna i en viss ordning. För att det ska bli rätt måste man börja i det övre vänstra hörnet och sluta i det nedre högra hörnet, och man får bara gå steg åt höger och nedåt. Om man lyckas besöka alla bokstavsrutorna på det här viset så bildas det hemliga meddelandet.

Paula har beslutat sig för att en gång för alla knäcka gåtan, och hitta meddelandet.

Skriv ett program som först läser in antalet rader (n) och kolumner (m) som rutnätet har. Läs sedan in n rader, som innehåller alla rutnätets rader. För alla testfall håller det att $1 \leq n, m \leq 6$.

Programmet ska sedan skriva ut det hemliga meddelandet som står på torget.

Poängsättning:

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

- (1) $n = 1$
- (2-3) Antalet bokstäver är $n + m - 1$
- (4-5) Inga särskilda begränsningar.

Körningsexempel 1

```
n ? 5
m ? 6
Rad 1 ? pa....
Rad 2 ? .....
Rad 3 ? .u.l..
Rad 4 ? .....a
Rad 5 ? .....
```

Svar: paula

p	a				
	u		l		
					a

Illustration av rutnätet i körningsexempel 1.

Körningsexempel 2

n ? 3

m ? 3

Rad 1 ? sn.

Rad 2 ? .a.

Rad 3 ? .ke

Svar: snake

UPPGIFT 4 – KLIPPA PROTEIN

Din kompis Björn har nyligen påbörjat ett stort kemiexperiment, med målet att skapa ett hemligt protein. Alla protein beskrivs med strängar av bokstäver från a-z, exempelvis `bcaa` och `hejsan`. Han behöver skapa ett flertal så kallade *rena* proteiner. Dessa är proteiner som består av kopior av en och samma bokstav. För att göra detta har han mikroskopiska saxar som kallas CRISPR. Med hjälp av dessa kan han ta bort upp till k bokstäver som ligger bredvid varandra. Med andra ord kan han välja ett intervall med upp till k bokstäver att klippa bort. Om ett protein delas i två delar när det klipps kommer dessa delar sedan att sättas ihop där de klipptes av.

Eftersom han är väldigt lat vill han ha din hjälp att skriva ett program som beräknar det minsta antalet gånger han behöver klippa ett visst protein för att det ska bli rent. Notera att proteinet **inte** får bli en tom sträng, eftersom det isåfall inte längre kan användas för kemi.

Programmet ska först läsa in Björns protein, som består av som mest 32 bokstäver. Sedan ska det läsa in talet k ($k \geq 1$).

Programmet ska sedan skriva ut det minsta antalet klippningar som krävs för att göra proteinet rent.

Poängsättning:

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

- (1) Strängen består endast av bokstäverna a och b.
- (2) $k = 1$
- (3-5) Inga särskilda begränsningar.

Körningsexempel 1

Björns protein ? exempelfall
k ? 4

Svar: 3

Förklaring: Ett möjligt sätt är först klippa bort bokstäverna `pe`. Vi får då strängen `exemlfall`.

Sedan kan vi klippa bort `fa` och får `exemlll`.

Till sist klippa vi bort `exem` och får `lll`.

Körningsexempel 2

Björns protein ? aabbabba
k ? 3

Svar: 2



Illustration av hur Björn kan klippa proteinet optimalt i körningsexempel 2.

UPPGIFT 5 – PICKLE CLICKER

För ungefär ett halvår sedan släpptes succéspelet *Pickle Clicker*. Efter att ha lagt otaliga timmar på spelet har Rasmus bestämt sig för att ta saker och ting till nästa nivå: han tänker speedrunna spelet.

Pickle Clicker går ut på att man samlar ihop så mycket som möjligt av valutan pickles. Målet i en speedrun är att köpa Megapickeln så fort som möjligt, som kostar t pickles. I spelet finns även n olika typer av byggnader. De olika typerna är numrerade från 1 till n , där den i :te typen producerar p_i pickles per sekund och kostar c_i pickles att bygga.

I början äger du en byggnad av typ 1. Varje sekund sker följande:

- (1) Varje byggnad du äger producerar pickles. En byggnad av typ i producerar p_i pickles.
- (2) Därefter har du alternativet att köpa en byggnad eller köpa Megapickeln. En byggnad av typ i kostar c_i pickles att bygga, och du måste ha råd med den för att få köpa den. Du får maximalt köpa en byggnad per sekund, men det är tillåtet att ha flera byggnader av samma typ.

Hjälp Rasmus bestämma hur han ska spela spelet så att han kan köpa Megapickeln på så få sekunder som möjligt. Se exemplet nedan för att få en tydligare bild av reglerna.

Programmet ska läsa in heltalen n och t ($1 \leq n \leq 6, 1 \leq t \leq 10^5$): antalet byggnader och kostnaden av Megapickeln.

Därefter ska programmet läsa in n rader: p_i och c_i , produktionshastigheten och kostnaden för vardera byggnad.

Programmet ska skriva ut ett heltal: hur lång tid det tar att vinna Pickle Clicker om man spelar optimalt.

Poängsättning:

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

- (1) $n = 1$
- (2) $t \leq 10$
- (3) $t \leq 1000$
- (4-5) Inga särskilda begränsningar.

Körningsexempel 1

Antal byggnader ? 2

Kostnad av Megapickeln ? 100

Produktionshastighet och pris för byggnad 1 ? 3 3

Produktionshastighet och pris för byggnad 2 ? 50 10

Svar: 5

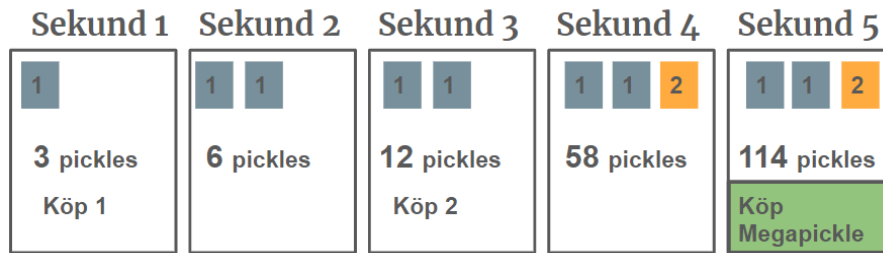


Illustration av vad man bör göra varje sekund för att lösa exempel 1 optimalt.

Körningsexempel 2

Antal byggnader ? 1

Kostnad av Megapickeln ? 100000

Produktionshastighet och pris för byggnad 1 ? 1 1000

Svar: 6179

UPPGIFT 6 – CHRISTIANS PIANO

Christian vill lära sig spela piano, men klarar bara av att spela med pekfingeret. Just nu håller han på att öva hårt för att lära sig ett stycke. Ett stycke beskrivs av en sekvens tangenter som ska tryckas i ordning. Exempelvis beskriver stycket `abdaa` att han ska trycka på tangenterna `a`, `b`, `d`, `a`, `a` i ordning. Du märker dock att han har svårt att röra pekfingeret snabbt nog mellan tangenterna. Märkvärdigt nog tar det honom alltid exakt en sekund att flytta fingret en tangent åt sidan, men noll sekunder att trycka ner en tangent. För att hjälpa honom tänker du ändra om ordningen på pianots tangenter så att totala tiden det tar att spela stycket blir så litet som möjligt.

Till exempel, om stycket han försöker lära sig är `abdaa` och pianot har ordningen `abd...` (resten av pianot visas inte då detta inte påverkar svaret) kommer det totalt ta 4 sekunder att spela stycket. Avstånden när han spelar stycket blir följande:

- `a` till `b` tar en sekund.
- `b` till `d` tar en sekund.
- `d` till `a` tar två sekunder.
- `a` till `a` tar noll sekunder.

Vilket innebär att det totalt tar fyra sekunder att spela stycket `abdaa` på ett piano med ordningen `abd...`.

Skriv ett program som läser in ett stycke och skriver ut den totala tiden det tar för Christian att spela stycket på pianot med bästa möjliga ordning på tangenter. Det är garanterat att stycket består av som mest 32 bokstäver och innehåller som mest 15 olika sorters bokstäver. Alla bokstäver i stycket är mellan `a-z`.

Poängsättning:

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

- (1) Stycket innehåller som mest 8 olika sorters bokstäver.
- (2-5) Inga särskilda begränsningar.

Körningsexempel 1

Stycket ? `abdaa`

Svar: 4

Körningsexempel 2

Stycket ? `abdaadbc`

Svar: 7

Förklaring: En möjlig optimal ordning på pianots tangenter är `adbc`.