Finaluppgifter i

Programmeringsolympiaden 2013

Fullständiga regler på

http://www.progolymp.se/elev/final

Kort lathund:

- Tävlingstid: sex timmar (avbrott för lunch tillåten)
- Ingen aktiv kommunikation gällande vare sig uppgifter eller programmering under tävlingstiden. Bara informationssökning.
- Inlämning via Kattis:
  - http://po.scrool.se/
  - Vid inskickning, ange problem-ID som står överst i uppgiftstexten
  - Inläsning via standard input
- Inlämning till lärare:
  - Döp källkodsfilerna efter problem-ID. Inkludera EXE-fil om Windows-miljö.
  - Inläsning antingen via standard input eller från filen *id*.dat, där *id* byts ut mot respektive problem-ID. Filen ska antas ligga i current directory.
- Om samma uppgift lämnas både på Kattis och till läraren gäller Kattis!

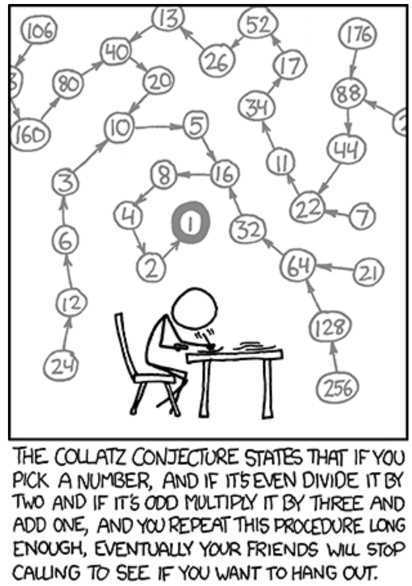
Hageltal

Problem ID: hageltal

Tänk dig att du skriver upp alla positiva heltal på ett oändligt stort papper. Från varje tal $n > 1$ ritar du nu en pil till talet

- $n/2$ om n är jämnt.
- $3n + 1$ om n är udda.

Ett avsnitt ur den graf som bildas visas i serierutan här intill. De talföljder man får genom att följa pilarna kallas ibland för *hageltal* eftersom de likt hagelkorn driver upp och ner längs tallinjen innan de slutligen faller ner till marken (talet 1). Det intressanta är att det fortfarande inte har bevisats att man verkligen alltid når talet 1, men det har verifierats för alla tal upp till 10^{19} så man *förmodar* det, vilket brukar kallas Collatz förmodan (conjecture).
Skriv ett program som, givet två olika heltal beräknar hur långt ifrån varandra (antal pilar, oavsett riktning) de är i grafen.



`http://xkcd.com/710/`

Indata

En rad med två olika heltal A och B, där $1 \leq A, B \leq 1000$.

Utdata

En rad med ett heltal, antal steg mellan A och B i grafen.

Sample Input 1	Sample Output 1
3 20	2

Sample Input 2	Sample Output 2
24 10	4

Sample Input 3	Sample Output 3
1 5	5

Sample Input 4	Sample Output 4
22 32	12

Sample Input 5	Sample Output 5
702 703	236

Bingo
Problem ID: bingo

Ture spelar en speciell sorts bingo som går till på följande sätt:

- Han har en kvadratisk bricka med NxN rutor, vardera innehållande ett heltal. Samma tal kan finnas i flera rutor.
- Spelledaren ropar ut ett godtyckligt heltal i taget. Om Ture har detta tal på sin bricka får han kryssa över det. Om han har talet i flera rutor måste han välja en enda ruta att kryssa över.
- När Ture har kryssat en full rad med N tal, antingen horisontellt, vertikalt eller diagonalt (se figur), så har han fått "bingo" och vinner en resa till BOI (Bingo Olympiad International).

Ture förlitar sig inte på turen utan har fuskat till sig den följd av tal som kommer att ropas ut. Skriv ett program som, givet Tures bricka samt följden av tal, beräknar efter hur många utrop Ture får bingo om han spelar optimalt.

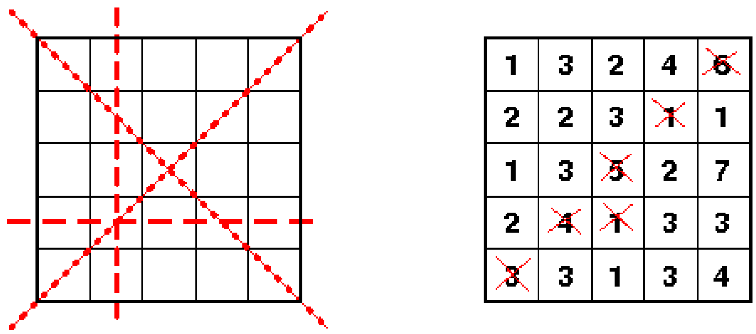


Figure 1: Vänster: De två diagonala raderna (prickade linjer) samt exempel på en horisontell och en vertikal rad (streckade linjer). Höger: Ett möjligt sätt att kryssa i det första exemplet.

Indata

Första raden innehåller brickans dimension N, där $1 \leq N \leq 100$, och antalet utrop K, där $1 \leq K \leq 10\,000$. Därefter följer N rader vardera innehållande N heltal i intervallet $1 \ldots 1000$. Detta är Tures bingobricka. Slutligen följer en rad med K heltal i samma intervall, de utropade talen i den ordning de ropas ut.

Utdata

Ett heltal: det minsta antalet utrop som behövs innan Ture får en full rad (horisontellt, vertikalt eller diagonalt). Det kommer aldrig krävas mer än K utrop.

Sample Input 1

```
5 10
1 3 2 4 6
2 2 3 1 1
1 3 5 2 7
2 4 1 3 3
3 3 1 3 4
4 1 3 5 1 6 7 2 8 3
```

Sample Output 1

```
6
```

Sample Input 2

```
8 30
15 20 20 13 18 13 14 20
17 11 13 20 9 10 8 19
19 12 9 8 16 19 9 4
20 17 3 9 1 14 14 9
12 20 19 5 16 5 19 17
3 4 17 8 5 14 18 17
9 14 16 13 13 9 13 1
13 7 8 3 19 19 5 7
2 5 1 1 20 4 18 6 20 2 19 17 15 1 6 14 17 7 12 10 10 17 18 3 3 12 13 9 18 17
```

Sample Output 2

```
28
```

Skolvägen
Problem ID: skolvagen

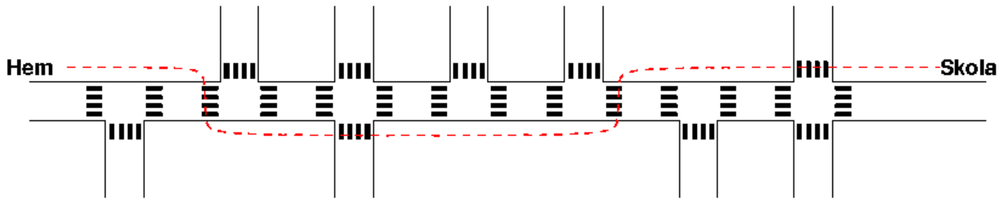


Figure 1: Den streckade linjen visar Cissis väg i första exemplet.

Cissi går från sitt hem till skolan längs en lång gata som går i väst-östlig riktning. På sin väg passerar hon ett antal korsningar där tvärgator utgår norrut (N), söderut (S) eller både norrut och söderut (B). Vid varje korsning finns övergångsställen på både tvärgator och huvudgata (se figuren ovan), och dessa måste givetvis följas.

Både hemmet och skolan ligger på norra sidan av gatan. Skriv ett program som hjälper Cissi att beräkna det minsta antalet gator hon måste korsa på sin väg till skolan.

Indata

En rad med högst 1000 bokstäver, som vardera är `N`, `S` eller `B`. Bokstäverna beskriver korsningarna i precis den ordning som Cissi passerar dem.

Delpoäng: För 40% av poängen är antalet bokstäver högst 20.

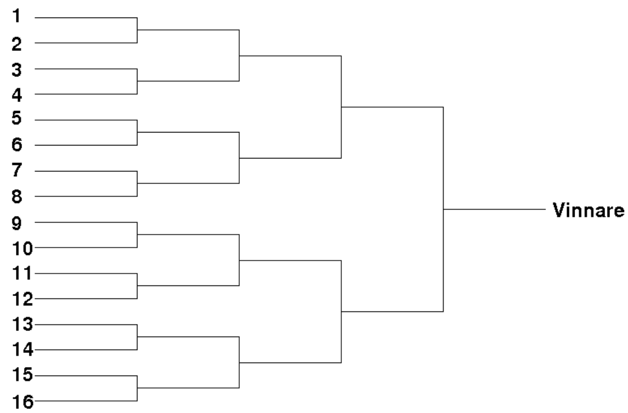
Utdata

En rad med ett heltal, det minsta antalet gator Cissi behöver korsa.

Sample Input 1	Sample Output 1
SNBNNSB	4

Sample Input 2	Sample Output 2
SBSNNBSNNSSSNNNB	8

Australian open
Problem ID: australian



Tennisturneringen Australian open är en *utslagsturnering* som kan beskrivas med ett spelschema enligt figuren ovan, där antalet *omgångar* n kan variera. Man startar alltså till vänster med 2^n spelare, numrerade från 1 till 2^n. Efter totalt $2^n - 1$ matcher har man utsett en vinnare.

Vi antar nu att varje spelare i har en "förmåga" x_i, och att när två spelare i och j möts, så vinner spelare i med sannolikheten

$$P(i \text{ besegrar } j) = \frac{1}{1 + \exp(x_j - x_i)}$$

och annars vinner spelare j eftersom en tennismatch inte kan sluta oavgjort ($\exp(x)$ betecknar exponentialfunktionen e^x). Formeln är vald så att $P(i \text{ besegrar } j) + P(j \text{ besegrar } i) = 1$.

Skriv ett program som, givet alla spelares förmågor, avgör vem som har störst chans att vinna turneringen och hur stor denna chans är.

Förklaring till första exemplet

Första matchen vinner spelare 1 med sannolikheten 0.549834 och spelare 2 med sannolikheten 0.450166. Andra matchen vinner spelare 3 med sannolikheten 0.689974 och spelare 4 med sannolikheten 0.310026. Dessa händelser är oberoende och därmed kan vi räkna ut sannolikheten för varje möjligt finalpar genom att multiplicera sannolikheterna med varandra. Det ger upphov till följande tabell:

	Sannolikhet att	**Sannolikhet att**	**Bidrag till total chans att spelaren vinner**			
Match	**matchen spelas**	**vinna matchen**	**1**	**2**	**3**	**4**
1–3	0.379371	0.524979–0.475021	0.199162	–	0.180209	–
1–4	0.170463	0.710950–0.289050	0.121190	–	–	0.049272
2–3	0.310603	0.475021–0.524979	–	0.147543	0.163060	–
2–4	0.139563	0.668188–0.331812	–	0.093254	–	0.046309
Summa	**1.000000**		**0.320352**	**0.240797**	**0.343269**	**0.095581**

Indata

På första raden står ett tal n, antal omgångar, där $1 \leq n \leq 10$. På andra raden står 2^n flyttal mellan 0 och 10, förmågan för varje spelare i nummerordning.

Utdata

Ett rad med ett heltal och ett flyttal: numret på den spelare som har störst chans att vinna turneringen samt sannolikheten att den spelaren vinner. Om flera spelare har störst chans kan du ange vilken som helst av dem. Sannolikheten ska skrivas ut med minst 6 decimaler.

Sample Input 1

```
2
2.0 1.8 1.9 1.1
```

Sample Output 1

```
3 0.34326946
```

Sample Input 2

```
4
9.99 9.53 9.93 8.77 8.57 9.55 8.12 9.41 8.20 9.43 8.91 8.63 8.88 9.83 8.69 8.40
```

Sample Output 2

```
14 0.20348623
```

Springoalla
Problem ID: springoalla

Springoalla älskar att löpträna. Totalt känner hon till n löpspår och hon vet exakt hur lång tid det tar för henne att springa det spåret och sedan tillbaka. Den första gången hon springer på ett nytt spår, så lär hon känna spåret lite bättre. Närmare bestämt lär hon sig var i spåret hon kommit halvvägs, och har då möjlighet att springa tillbaka efter halva spåret. Då blir löptiden halverad. T.ex. kan hon springa ett halvt 20-minutersspår på 10 minuter, men bara efter att hon redan sprungit hela spåret en gång.

Springoalla vill löpträna i minst t minuter men hon är också noggrann med att inte träna för länge. Givet tiderna för varje spår, beräkna hur lång tid t_s hon minst måste springa. Talet t_s ska alltså vara så litet som möjligt men uppfylla $t_s \geq t$. Om det finns flera sätt att springa t_s minuter vill Springoalla springa så litet antal sträckor n_s som möjligt, där man räknar varje gång hon springer bort från utgångspunkten som en sträcka, oavsett om det är ett helt eller halvt spår. Finns det flera lösningar med samma t_s och n_s kan du ange vilken som helst av dem.

Indata

På första raden står två heltal n och t, där $1 \leq n \leq 1000$ är antalet löpbanor och $1 \leq t \leq 100\,000$ är tiden som Springoalla vill löpträna. På andra raden står n stycken heltal l_i, där $1 \leq l_i \leq 40\,000$ kommer vara ett *jämnt* heltal och är antalet minuter det tar att springa löpspår i. Talet t behöver däremot inte vara jämnt.

Utdata

Första utdataraden ska innehålla de två heltalen t_s och n_s: den tid Springoalla måste springa respektive hur många sträckor hon totalt springer. Därefter ska en rad skrivas med n heltal, där det i:te heltalet anger hur många minuter Springoalla sprang på spår i.

Sample Input 1	Sample Output 1
3 23 10 8 14	23 3 15 8 0

Sample Input 2	Sample Output 2
3 23 8 12 14	24 2 0 24 0

Sample Input 3	Sample Output 3
1 3 2	3 2 3

Sample Input 4	Sample Output 4
1 7 4	8 2 8

Storstad
Problem ID: storstad

Deltagarna i den svenska programmeringsolympiaden älskar geometri, och det med rätta. Den största favoriten bland geometrierna är den så kallade *manhattangeometrin*. Manhattanavståndet mellan två punkter **P** och **Q** i ett regelbundet rutnät beskriver hur många punkter man måste passera för att ta sig från **P** till **Q**, givet att man bara får gå horisontellt och vertikalt.

Om **P** $= (x_1, y_1)$ och **Q** $= (x_2, y_2)$, så beräknas manhattanavståndet mellan punkterna som $|x_1 - x_2| + |y_1 - y_2|$. $|a|$ betecknar absolutbeloppet för talet a, dvs vi ignorerar minustecknet om talet är negativt.

Nu är det dags för dig att visa hur mycket du älskar geometri. Du ska planera din flytt till en storstad där manhattanavstånd gäller. Du har N stycken vänner som bor i storstaden, men du har bara M lägenheter att välja på. Din uppgift är att lista ut hur du ska välja lägenhet så att summan av manhattanavstånden till alla dina N vänner från din lägenhet är *så liten som möjligt*.

Delpoäng

På det här problemet kan du samla poäng trots att du inte löser problemet helt och hållet.

- För 40% av poängen gäller att $1 \leq N \leq 10^4$ och $1 \leq M \leq 100$.
- För full poäng måste din lösning hantera $1 \leq N \leq 10^5$ och $1 \leq M \leq 10^5$.

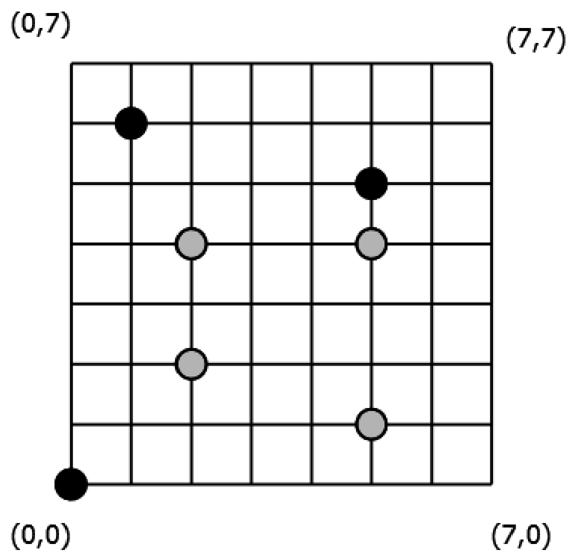


Figure 1: De svarta prickarna representerar lägenheter vi kan välja på i storstaden, och de grå prickarna markerar var våra vänner bor. Om vi väljer lägenheten i (0,0) så blir summan av avstånden till vännerna 4+6+6+9=25. Väljer vi lägenheten i (1,6) så blir summan av avstånden 3+5+6+9=23. Väljer vi den sista lägenheten så blir summan av avstånden 1+4+6+4=15, och att välja lägenheten i (5,5) är då optimalt.

Indata

På första raden i indata finns två heltal N och M, separerade av ett mellanslag. Sedan följer N rader med två par av heltal $-10000 \leq x_i \leq 10000$ och $-10000 \leq y_i \leq 10000$, separerade av mellanslag. Dessa beskriver på vilka korsningar i staden som dina vänner bor. Slutligen följer M rader med par av tal $-10000 \leq x_i \leq 10000$ och $-10000 \leq y_i \leq 10000$, separerade av mellanslag, som beskriver på vilka korsningar i staden du kan välja att bo. Notera att det kan finnas flervåningshus i storstaden (en punkt kan förekomma flera gånger i indata).

Utdata

Ditt program ska skriva ut ett enda tal på en rad: den minsta möjliga summan av manhattanavstånden till dina N vänner från din lägenhet.

Sample Input 1	Sample Output 1
4 3 2 2 5 4 2 4 5 1 5 5 0 0 1 6	15

Sample Input 2	Sample Output 2
2 2 0 0 0 2 2 0 2 2	6

Sample Input 3	Sample Output 3
1 1 1 1 1 1	0

Räkneord
Problem ID: rakneord

Monika skriver ner alla räkneord mellan 1 och N. Sedan sorterar hon orden alfabetiskt och konkatenerar dem till en enda lång sträng. Skriv ett program som hittar bokstav nummer K, $K+1$ och $K+2$ i Monikas sträng.

För att undvika problem med teckenkodning byter vi ut å och ä mot a. Vidare använder vi formerna etthundra, ettusen (observera stavningen), enmiljon och enmiljard, men bara när siffran 1 står *ensam* före mängdordet (ej inräknat tidigare mängdord). När siffran 1 däremot står sist i ett förled som betecknar *flera* tusen, miljoner eller miljarder så skriver vi för enkelhets skull alltid "en": tjugoentusen, etthundraenmiljoner, tvahundraattioenmiljarder etc. Allra sist i ett räkneord skriver vi dock alltid siffran 1 som "ett".

En tabell med de grundläggande räkneorden och några exempel på längre ord bringar förhoppningsvis klarhet i reglerna:

Tal	Räkneord	Tal	Räkneord
1	ett	10	tio
2	tva	20	tjugo
3	tre	30	trettio
4	fyra	40	fyrtio
5	fem	50	femtio
6	sex	60	sextio
7	sju	70	sjuttio
8	atta	80	attio
9	nio	90	nittio
11	elva	100	etthundra
12	tolv	198	etthundranittioatta
13	tretton	201	tvahundraett
14	fjorton	1121	ettusenetthundratjugoett
15	femton	581 743	femhundraattioentusensjuhundrafyrtiotre
16	sexton	51 101 001	femtioenmiljoneretthundraentusenett
17	sjutton	162 500 020	etthundrasextiotvamiljonerfemhundratusentjugo
18	arton	1 002 001 004	enmiljardtvamiljonerettusenfyra
19	nitton	91 011 091 000	nittioenmiljarderelvamiljonernittioentusen

Och som en ytterligare kontrollmöjlighet ger vi längden på Monikas sträng för några olika N:

N	Strängens längd
999	16260
9999	235600
99999	2908000
999999	37425000
9999999	472250000
99999999	5319500000
999999999	61585000000
9999999999	722850000000
99999999999	7834500000000
999999999999	86744000000000

Indata

En rad med två heltal N och K, där $1 \leq N \leq 999\,999\,999\,999$ och $1 \leq K \leq L - 2$, där L är längden på den bildade strängen.

Delpoäng: För 30% av poängen är $N \leq 500\,000$, medan för resten av poängen är $N \geq 100\,000\,000$. För 30% av poängen är $N = 999\,999\,999\,999$.

Utdata

En rad med tre bokstäver utan åtskiljande blanksteg: bokstav nummer K, $K+1$ och $K+2$ i strängen.

Förklaring till första exemplet

Monikas sträng, skriven med 40 bokstäver per rad, är

```
artonattaelvaettfemfemtonfjortonfyranion
ittonsexsextonsjusjuttontiotjugotjugoett
tjugofemtjugofyratjugosextjugotretjugotv
atolvtretrettontva
```

Sample Input 1	Sample Output 1
26 120	vat

Sample Input 2	Sample Output 2
105 508	ase

Sample Input 3	Sample Output 3
426760 11111111	atr