

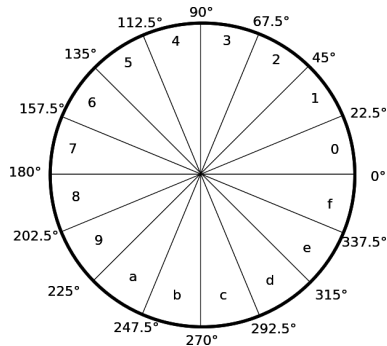
Problem A

Marsmeddelanden

Problem ID: mars

I filmen *The Martian* blir astronaut Mark Watney fast på Mars. Jorden kommunicerar med Mark genom att rotera kameran på en gammal rymdsond.

Sonden pekar på en hexadecimal siffra, d.v.s. 0–9 eller a–f, och Mark bildar sedan bokstäver och andra tecken genom att konvertera par av dessa siffror enligt teckenkodningen ASCII. För att tyda vilken siffra sonden pekar på har han delat in en hel cirkel om 360 grader i 16 lika stora bitar om 22.5 grader vardera.



Som ett exempel representerar vinkelparet $(100^\circ, 30^\circ)$ det hexadecimala talet 41, vilket är talet 65 decimalt. Enligt ASCII¹ representerar detta tecknet A.

Nu har Mark tröttnat lite på att skriva ner dessa siffror manuellt, och har bett dig om hjälp med att avkoda meddelandet som han får. Du kommer få sekvensen av de vinklar som sonden pekar på, och ska skriva ut meddelandet.

Input

Den första raden innehåller ett heltal $2 \leq N \leq 50\,000$, antalet vinklar du ska läsa in. N garanteras vara ett jämnt tal. De nästa N raderna innehåller vinklarna, en vinkel per rad. Vinkeln ges som ett flyttal i grader i intervallet $[0, 360)$. Det är garanterat att sonden aldrig pekar på någon vinkel som ligger precis mellan två hexadecimala siffror. Det är också garanterat att under avkodningen till ASCII kommer enbart tecken med decimala värden 32–126 ges.

Output

Du ska skriva ut en enda rad med de tecken som meddelandet avkodas till.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Begränsningar
1	46	alla vinklar är heltal
2	54	

¹https://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

Sample Input 1

54
91
118
157
23
167
66
158
100
157
189
47
21
172
98
139
353
59
10
139
314
146
23
160
48
162
70
63
316
55
21
94
95
144
353
53
11
166
203
147
339
158
115
46
3
157
83
149
339
160
18
177
218
69
339

Sample Output 1

Earth to mars. Do you copy?

Problem B

Gratis mat

Problem ID: gratismat

Hsara älskar gratis mat.

Ikväll är Hsara på restaurang med jobbet, och Hsara sitter och analyserar möjligheterna för att få middagen betald. Precis en person kommer att betala för hela middagen (allas mat), men inte vilken person som helst. Det är endast tillåtet att betala för middagen om ingen överordnad (eventuellt indirekt, t.ex. din chefs chef) i organisationen befinner sig vid bordet.

Organisationen består av N personer numrerade från 1 till N . Varje person, förutom organisationens VD, har exakt en chef. Givet varje persons chef och en lista över personer som är närvarande vid bordet, avgör hur många som skulle kunna betala för middagen.

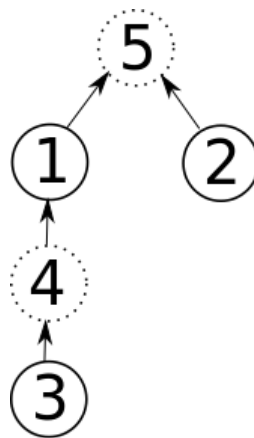


Figure B.1: Illustration av hur organisationen i det första exemplet ser ut.

Input

Den första raden i indatan innehåller två heltal, N - antalet personer i organisationen och $1 \leq M \leq N$ - antalet personer vid bordet.

Den andra raden innehåller N heltal. Det i :te talet anger vem som är chef till person i . 0 innebär att personen inte har en chef och därmed är VD för organisationen. Det finns precis en person som är VD.

Den tredje raden innehåller M heltal som anger vilka som sitter vid bordet.

Output

Skriv ut ett heltal - antalet personer vid bordet som skulle kunna betala för middagen.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Begränsningar
1	49	$2 \leq N \leq 2\,000$
2	51	$2 \leq N \leq 100\,000$.

Förklaring

I det första exemplet kan personerna 1 och 2 betala. Person 3 kan inte betala eftersom person 1 sitter vid bordet.

I det andra exemplet är person 5 VD och sitter vid bordet. Därför kan bara person 5 betala.

Sample Input 1

```
5 3
5 5 4 1 0
1 2 3
```

Sample Output 1

```
2
```

Sample Input 2

```
6 5
5 4 4 5 0 1
5 1 3 2 4
```

Sample Output 2

```
1
```

Problem C

Labplanering

Problem ID: labplanering

Ah, universitetet. En kunskapens högborg, utbildningens Mecka, och byråkratins kärna.

Wille och Kashi sitter just nu och ska redovisa en datorlaboration i en kurs. Totalt är det N grupper som sitter i datasalen och väntar på att få redovisa. Grupp i ska redovisa totalt m_i olika moment, där moment j tar $a_{i,j}$ minuter.

För att det ska vara rättvist måste alla grupper vänta jättelänge på sina redovisningar. Det finns nämligen bara en lärare som tar redovisningar, och att låta varje grupp redovisa samtliga av sina moment i ett enda svep skulle ju innebära att en grupp aldrig måste stanna kvar längre än nödvändigt! Därför tar läraren emot redovisning av ett moment i taget i en ganska godtycklig ordning...

Kashi har fått nog. Hon vill hem och spela Pokémon, och klagar på den ganska ineffektiva redovisningsordningen högljutt. För att demonstrera just hur ineffektivt detta system är tänker hon titta på vad den längsta möjliga totala väntetiden är, och visa hur nära lärarens system är till detta.

Antag att en grupp börjar redovisa sitt första moment vid tid a , och blir färdig med redovisningen av sitt sista moment vid tid b . Då är väntetiden för denna grupp $b - a$. Den totala väntetiden är väntetiden summerad över alla grupper som redovisar.

Ordningen i vilken en grupps moment ska redovisas måste vara som i indata.

Input

Den första raden innehåller talet $N \geq 1$.

Därefter följer N rader, som vardera innehåller först talet $m_i \geq 1$, och sedan m_i heltal $a_{i,j}$, alla mellan 1 och 60.

Output

Du ska skriva ut ett enda tal: den längsta möjliga totala tid du kan få studenterna att stanna om du planerar schemat optimalt.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Summan av alla m_i kommer att vara högst 5 000 i grupperna 1 till och med 4.

Grupp	Poängvärde	Begränsningar
1	30	summan av m_i är högst 10
2	17	alla moment är lika långa
3	13	alla grupper har precis 2 moment att redovisa
4	18	inga ytterligare restriktioner
5	22	summan av m_i kan bli uppemot 100 000

Förklaring

I det givna exemplet ska tre gruppers labbar redovisas: den första gruppens moment tar 5 resp. 15 minuter, den andra 10 resp. 20, och den tredje gruppens moment tar en hel timme.

Om vi lägger redovisningarna i ordningen 5, 10, 60, 20, 15 så kommer den tredje gruppen kunna göra sin redovisning på 60 minuter. Den andra gruppen kommer då att förutom sin egen labb behöva vänta igenom hela den tredjes redovisning, och därmed ta $10 + 60 + 20 = 90$ minuter på sig. På samma sätt kommer den första gruppen behöva vänta $5 + 10 + 60 + 20 + 15 = 110$ minuter. Summan av tiderna blir $60 + 90 + 110 = 260$ minuter.

(Notera att ordningen 10, 15, 60, 20, 5 hade resulterat i en värre tidssumma på 265, men detta tillåts inte: grupp 1 får inte redovisa sin andra labb före sin första.)

Sample Input 1

```
3
2 5 15
2 10 20
1 60
```

Sample Output 1

```
260
```

Problem D

Frågetävling

Problem ID: fragetavling

I frågetävlingen ProgrammeringsQuiz finns det totalt N frågor, fördelade över M olika kategorier (t.ex. algoritmteori, kompilatorkonstruktion eller Sven-kunskap).

Frågorna är värda olika mycket poäng. Dessutom får du en bonus B när du löser samtliga problem i en viss kategori. Simone har varit med i Programmeringsolympiaden sen åttonde klass, så hon kommer kunna svara rätt på alla frågor.

Tyvärr går tävlingen på tid. Trots att hon aldrig svarar fel kommer hon bara hinna svara på K av frågorna. Vilken poäng kan Simone maximalt uppnå?

Input

Den första raden innehåller de fyra heltalen $1 \leq N \leq 1000$, $1 \leq M \leq N$, $1 \leq K \leq N$, $1 \leq B \leq 100\,000$. De följande N raderna innehåller två tal var: poängen för frågan (ett heltal mellan 1 och 1 000) och vilken kategori den ingår i (mellan 1 och M). Det är garanterat att alla kategorier kommer att innehålla minst en fråga.

Output

Du ska skriva ut en enda rad med den maximala poängen du kan uppnå.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Begränsningar
1	16	$N \leq 100, M = 1$
2	29	$M \leq 5$
3	20	alla frågor ger lika mycket poäng
4	35	

Förklaring

I det första exemplet svarar vi på båda frågorna i kategori 1 ($300 + 400 = 700$ poäng) och den enda frågan i kategori 2 (200 poäng). Eftersom dessa var de enda frågorna i de två kategorierna får vi även två stycken kategoribonusar, vilket ger totalt $200 + 700 + 2 \cdot 1000 = 2900$ poäng.

Sample Input 1

Sample Input 1	Sample Output 1
5 3 3 1000 300 1 400 1 200 2 200 3 300 3	2900

Sample Input 2

```
5 3 3 1
300 1
400 1
200 2
300 3
200 3
```

Sample Output 2

```
1001
```

Problem E

Vvvvvv

Problem ID: vvvvvv

Plattformspel är en gammal dataspelgenre där du rör dig på plattformar som hänger i luften. Vvvvvv är en sorts plattformspel där du inte kan hoppa, men istället kan växla mellan vanlig gravitation, som gör att du faller nedåt, och antigravitation, som gör att du faller uppåt. Förutom gravitationsomkopplaren (G) så finns bara två knappar i spelet, gå till höger (H) och gå till vänster (V).

I den här uppgiften ska du hitta en sekvens av knapptryckningar som tar dig genom en labyrinthliknande uppsättning av plattformar och väggar, från en startposition till en slutposition. För att få full poäng ska sekvensen dessutom vara så kort som möjligt.

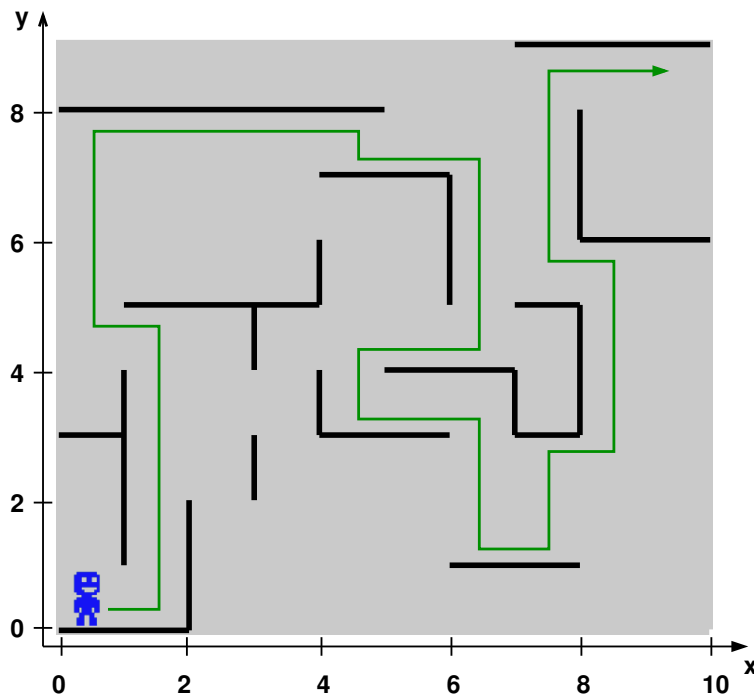


Figure E.1: Vägen som ges av knapptryckningssekvensen i det andra exemplet.

Input

Den första raden i indatan innehåller tre heltal W , H , och N : spelplanens bredd, spelplanens höjd samt antalet linjesegment. Därefter följer N rader med fyra heltal x_1 , y_1 , x_2 , y_2 på varje rad, där $0 \leq x_1, x_2 \leq W$ och $0 \leq y_1, y_2 \leq H$. Varje rad beskriver ett linjesegment med ändpunkterna (x_1, y_1) samt (x_2, y_2) . Varje linjesegment är antingen vågrätt eller lodrätt, d.v.s. antingen gäller $y_1 = y_2$ eller $x_1 = x_2$. Heltalskoordinaterna kan sägas dela in den rektangulära spelplanen i ett rutnät, i vilket knappen V förflyttar dig en ruta åt vänster medan H förflyttar dig en ruta åt höger. Om du inte står på en plattform efter förflyttningen så faller du (uppåt eller nedåt) till närmsta plattform, och det finns ingen möjlighet att röra dig i sidled under fallets gång. Om du faller eller går ut ur spelplanen förlorar du spelet. Detsamma gäller om du försöker gå genom en vägg.

Startpositionen är rutan i nedre vänstra hörnet närmast punkten $(0, 0)$ med gravitationen riktad neråt. Slutpositionen är rutan i övre högra hörnet, närmast punkten (W, H) . Det kommer alltid finnas ett linjesegment under startpositionen samt över slutpositionen.

Output

Om det finns en lösning ska programmet skriva ut en sträng innehållande en bokstav för varje knapptryckning, valda bland H, V och G. Sekvensen ska ta dig från startpositionen till slutpositionen och måste vara kortare än 10 000 tecken. Du kan anlända till slutpositionen med valfri riktning på gravitationen, men du måste stå stilla, alltså ha "fallit färdigt" (se tredje exemplet nedan).

Om det inte finns en lösning ska programmet skriva ut strängen "Inte" (utan citationstecken).

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Begränsningar	Krav
1	20	$W, H \leq 10, N \leq 25$, lösning finns med högst 10 drag	sekvensen behöver inte vara kortast möjlig.
2	15	$W, H \leq 20, N \leq 200$, lösning finns där man inte behöver använda V	sekvensen behöver inte vara kortast möjlig.
3	15	$W, H \leq 50, N \leq 2000$	sekvensen behöver inte vara kortast möjlig.
4	20	$W, H \leq 10, N \leq 50$	sekvensen ska vara kortast möjlig.
5	30	$W, H \leq 100, N \leq 5000$	sekvensen ska vara kortast möjlig.

Sample Input 1

```
4 5 9
2 5 4 5
4 5 4 1
4 4 3 4
3 0 4 0
0 0 2 0
0 3 2 3
2 2 2 3
1 1 1 2
1 1 3 1
```

Sample Output 1

```
HGHHVH
```

Sample Input 2

```
10 9 22
0 0 2 0
2 0 2 2
1 1 1 4
0 3 1 3
1 5 4 5
3 2 3 3
3 4 3 5
4 5 4 6
4 4 4 3
4 3 6 3
6 1 8 1
5 4 7 4
7 4 7 3
7 3 8 3
8 3 8 5
8 5 7 5
6 5 6 7
6 7 4 7
0 8 5 8
8 8 8 6
8 6 10 6
7 9 10 9
```

Sample Output 2

```
HGVHHHHGHVHVHHHHGHVHH
```

Sample Input 3

```
2 1 2
0 0 1 0
1 1 2 1
```

Sample Output 3

```
Inte
```

Problem F

Godis

Problem ID: godis

Det är lördag, och Ann Britt-Caroline ska handla godis. Hon har identifierat ett antal olika godispåsar hon kan tänka sig att köpa.

Varje påse innehåller ett antal godisbitar av olika sorter. Det finns 10 olika sorters vanligt godis (dessa är numrerade 1...10), och 10 sorter av en ny typ av anti-godis (numrerade $-1... -10$). Det händer sig nu att godisbitar av sort n och av sort $-n$ inte går särskilt väl ihop - de annihileras när de kommer i kontakt med varandra. I övrigt har dock anti-godiset samma konsistens och smak som vanligt godis.

När Ann Britt-Caroline köpt alla godispåsar hon vill ha så blandar hon ihop påsarna i en stor skål, så att samtliga par av godis/anti-godisar tar ut varandra (hon blandar godisskålen grundligt). Hur många godisbitar kan Ann Britt-Caroline ha kvar i slutet (efter att godis/anti-godis tagit ut varandra), om hon väljer godispåsar på bästa sätt? Observera att hon bara kan köpa en enda godispåse av varje sort. Ignorera eventuella pengabekymmer - hennes föräldrar bjuder.

Input

Den första raden i indatan innehåller ett heltal $1 \leq N \leq 1000$ - antalet godispåsar.

De följande N raderna beskriver varje godispåse. Först på raden finns ett tal $1 \leq k \leq 10$, som specificerar antalet olika sorters godis i påsen. Sedan följer k par av tal s n , vilket betyder att det finns n godisar av sort s . Varje sort nämns högst en gång per godispåse, och sorterna s och $-s$ kan av naturliga skäl inte förekomma tillsammans i samma påse.

Alla n kommer att uppfylla $1 \leq n \leq 1000$.

Output

Skriv ut ett tal: den största mängden godis Ann Britt-Caroline kan ha i slutändan.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Begränsningar
1	25	$N \leq 15$
2	25	$N \leq 1000$. Varje påse innehåller bara en sorts godis.
3	50	$N \leq 1000$

Förklaring till exemplet

Den största mängden får Ann Britt-Caroline om hon köper de två första påsarna. Då kommer en godis av sort 1 och en godis av sort -1 elimineras, och kvar blir två godisar kvar av sort 1, och fem av sort -2.

Sample Input 1	Sample Output 1
3 1 1 3 2 -1 1 -2 5 2 2 2 -3 1	7

Problem G

Solitärschack

Problem ID: solitarschack

Mårten är bra på schack. Lite för bra, enligt vissa. Till exempel Johan. Mårten utmanar ofta Johan på schack, vilket leder till att Johan förlorar på schack. Istället för att erkänna sig besegrad gör Johan vad han brukar göra när han förlorar på något; han ändrar reglerna.

Denna gång har han uppfunnit ett nytt slags enspelarschack som han kallar för *Solitärschack*. Spelet körs på ett 6×6 bräde med pjäser. Pjäserna finns i tre olika färger - brons, silver, guld - och åtta olika typer: 1, 2, 3, 4, torn, lopare, dam och springare. I början är brädet helt och hållet fyllt med bronsfärgade pjäser.

Varje drag i spelet består av att Johan plockar bort en pjäs från brädet. Om pjäsen som plockas bort är en bronspjäs, ersätts denna med en ny silverpjäs av slumpmässigt utvald typ. Om den bortplockade pjäsen är en silverpjäs ersätts den med en ny guldpjäs av slumpmässigt utvald typ. Om pjäsen som Johan tar bort är en guldpjäs så lämnas rutan tom. Spelets grundläggande mål är att utföra så många drag som möjligt.

I det första draget får Johan välja fritt vilken pjäs som ska plockas bort. Vid efterföljande drag begränsas vilka pjäser som får plockas bort av ett antal regler, som beror av vilken typ och position den senast bortplockade pjäsen hade. Johan får fortsätta göra drag så länge han kan och vill.

Låt (r, c) vara rad och kolumn för den pjäs som senast blev borttagen, och (r', c') rad och kolumn för en pjäs du vill ta bort härnäst. Raderna och kolumnerna ett-indexeras. Huruvida detta drag är tillåtet avgörs av den av de nedanstående reglerna som matchar typen för den senast borttagna pjäsen:

- 1** nästa pjäs måste vara exakt 1 steg (horisontellt, vertikalt eller diagonalt) från denna pjäs. Mer formellt måste något av följande gälla:

$$|r - r'| = 1, |c - c'| = 0$$

$$|r - r'| = 0, |c - c'| = 1$$

$$|r - r'| = 1, |c - c'| = 1$$

- 2** nästa pjäs måste vara exakt 2 steg (horisontellt, vertikalt eller diagonalt) från denna pjäs. Mer formellt måste något av följande gälla:

$$|r - r'| = 2, |c - c'| = 0$$

$$|r - r'| = 0, |c - c'| = 2$$

$$|r - r'| = 2, |c - c'| = 2$$

- 3** nästa pjäs måste vara exakt 3 steg (horisontellt, vertikalt eller diagonalt) från denna pjäs. Mer formellt måste något av följande gälla:

$$|r - r'| = 3, |c - c'| = 0$$

$$|r - r'| = 0, |c - c'| = 3$$

$$|r - r'| = 3, |c - c'| = 3$$

- 4** nästa pjäs måste vara exakt 4 steg (horisontellt, vertikalt eller diagonalt) från denna pjäs. Mer formellt måste något av följande gälla:

$$|r - r'| = 4, |c - c'| = 0$$

$$|r - r'| = 0, |c - c'| = 4$$

$$|r - r'| = 4, |c - c'| = 4$$

torn nästa pjäs måste ligga antingen rakt uppåt, nedåt, till vänster eller till höger om denna pjäs, förflyttad ända bort mot kanten. Mer formellt måste något av följande gälla:

$$r' \in \{1, 6\}, |c - c'| = 0$$

$$|r - r'| = 0, c' \in \{1, 6\}$$

lopare nästa pjäs måste vara längs med brädets sidor, på samma diagonal som denna pjäs. Mer formellt måste något av följande gälla:

$$|r - r'| = |c - c'|, r' \in \{1, 6\}$$

$$|r - r'| = |c - c'|, c' \in \{1, 6\}$$

dam nästa pjäs måste vara placerad som om denna pjäs var antingen **torn** eller **lopare**. Mer formellt måste något av följande gälla:

$$r' \in \{1, 6\}, |c - c'| = 0$$

$$|r - r'| = 0, c' \in \{1, 6\}$$

$$|r - r'| = |c - c'|, r' \in \{1, 6\}$$

$$|r - r'| = |c - c'|, c' \in \{1, 6\}$$

springare nästa pjäs måste vara belägen antingen på raden ovan/under och 2 kolumner till vänster/höger, eller på kolumnen till vänster/höger och 2 rader ovan/under. Mer formellt måste något av följande gälla:

$$|r - r'| = 2, |c - c'| = 1$$

$$|r - r'| = 1, |c - c'| = 2$$

I samtliga fall är det inte tillåtet att ha $(r', c') = (r, c)$.

Poängsättningen är som följer. För varje ruta när spelet är över får man följande poäng per ruta:

brons 0 poäng

silver 1 poäng

guld 2 poäng

tom 3 poäng

Man kan också få bonuspoäng under spelet. Följande bonusar finns:

- Om du tar bort $N \geq 2$ pjäser i följd av samma typ får du $2N$ poäng.
- Om du tar bort pjäserna 1, 2, 3, 4 i följd i ordning eller omvänd ordning får du 12 poäng.
- Om du tar bort pjäserna 1, 2, 3, 4 i följd i en annan ordning får du 8 poäng.
- Om du tar bort pjäserna *torn*, *lopare*, *dam*, *springare* i följd i någon ordning får du 8 poäng.
- Att ta bort siffrorna 1, 2, 3, 4 i någon ordning kallas för ett *sifferset*. Att ta bort pjäserna *torn*, *lopare*, *dam*, *springare* i någon ordning kallas för ett *pjässet*.

Om du tar bort $K \geq 2$ stycken sifferset och pjässet i *alternerande ordning*, d.v.s antingen sifferset - pjässet - sifferset - ... eller pjässet - sifferset - pjässet - ..., så får du ytterligare $8K$ poäng.

Bonusar av en och samma sort kan inte överlappa, och de delas ut girigt. T.ex. skulle sekvensen av borttagna pjäser 4 1 2 3 4 *dam torn springare lopare* ge $8 + 8 = 16$ bonuspoäng: 8 för sekvensen 4 1 2 3, inga för sekvensen 1 2 3 4 eftersom den överlappar med den tidigare, och 8 för sekvensen *dam torn springare lopare*. Inga ytterligare poäng för alternerande pjässet ges, eftersom vi inte fått poäng för sekvensen 1 2 3 4. Sekvensen 1 1 2 3 4 ger $4 + 12 = 16$ poäng: 1 1 och 1 2 3 4 är olika typer av bonusar, så det är okej att de överlappar.

Det visade sig dock vara svårare att få bra poäng i Solitärshack än att slå Mårten på vanlig schack... Hjälp Johan att få så bra poäng på Solitärshack som möjligt!

Input

Indatat börjar med 6 rader, med 6 blankstegsseparerade strängar vardera. Dessa anger vilka de ursprungliga bronspjäserna är.

Interaktion

Detta är ett interaktivt problem. För att göra ett drag skriver du ut två heltal $r \ c$, den rad och kolumn som innehåller den pjäs du vill ta bort.

Du ska sedan läsa in en rad - namnet på den pjäsen som ersatte den du tog bort. Om du just tog bort en guldpjäs kommer raden att innehålla ordet `blank`.

När du inte vill göra fler drag ska du skriva ut en rad `0 0` och avsluta programmet.

Glöm inte att flusha standard out när du printat! I Java gör du detta med `System.out.flush()`, i C med `fflush(stdout)`, i C++ med `flush(cout)`, i C# med `Console.Out.Flush()` och i Python med `sys.stdout.flush()`.

Exempel

Domarens utskrifter är i fetstil.

```
1 1 1 1 1 1
1 1 1 1 1 1
1 1 2 1 1 torn
1 1 3 4 1 dam
1 1 1 1 1 1
1 1 1 1 1 1
6 6
dam
5 6
springare
6 6
torn
1 1
lopare
0 0
```

I det här exemplet tog du först bort ettan vid position (6, 6), sedan den vid (5, 6), sedan damen som dykt upp vid (6, 6), och slutligen ettan vid position (1, 1). Därefter gav du upp i tristess.

Notera att dragen du gör är utifrån vilken sorts pjäs som *just togs bort*, inte vilken som just visades. Efter borttagandet av damen hade du också kunnat ta bort en pjäs på position (6, 1) eller (1, 6), dock ej t.ex. position (4, 1) eftersom man vid dam(/torn/lopare)drag måste flytta sig ända till en kant.

Poängsättning

Antag att på ett testfall värt P poäng så får ditt program X poäng, och det bästa programmet Y poäng. Då får du $P \times \frac{X}{Y}$ poäng. Innan tävlingsslut kommer alla lösningar rättas med $Y = 200$. Efter tävlingsslut rättas detta om med de bästa deltagarlösningarna på varje testfall.

Testfallen är indelade i ett antal grupper.

Grupp	Poängvärde	Begränsningar
1	4	Endast pjäsen springare förekommer.
2	4	Endast pjäsen 1 förekommer.
3	6	Endast pjäserna 1 och 2 förekommer.
4	9	Endast pjäserna torn, lopare, dam förekommer.
5	24	Endast pjäserna 1, 2, 3, 4 förekommer.
6	53	Alla pjäser kan förekomma.