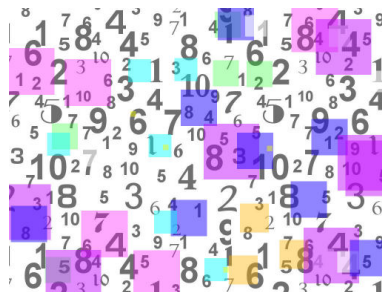


## Unike tall

En morsom lek å leke når man er mange samlet er “laveste unike tall”. I dette spiller så skriver hver deltager ned et positivt heltall på en lapp. Vinneren er den som har skrevet det laveste tallet som ingen andre har skrevet.

Du har fullført en runde med denne leken, men er nå usikker på hvilket tall som endte opp med å bli vinnertallet. Du vil derfor ha et program som lar deg finne ut hva det laveste tallet som kun står på én av lappene er.



## Input

Første linje inneholder et heltall  $N$ : antall lapper med tall.

Deretter følger  $N$  linjer, hver med et heltall  $X_i$  - tallet på den  $i$ 'te lappen.

## Output

Et tall  $V$ , vinnertallet i denne runden. Dersom ingen av tallene er unike skal du skrive ut  $-1$  i stedet.

## Begrensninger

$$2 \leq N \leq 50\,000$$

$$1 \leq X_i \leq 1\,000\,000\,000$$

**Tidsbegrensning:** 1 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	20	$N \leq 5\,000$ ; Dersom et vinnertall eksisterer så er det 10 eller mindre.
Gruppe 2	35	$N \leq 5\,000$
Gruppe 3	45	Ingen andre begrensninger

## Eksempler

Input	Output	Kommentarer
7 4 3 8 20 4 4 3	8	Det laveste tallet som har blitt skrevet ned er 3, men siden 2 personer skrev dette tallet er det ikke unikt. Det laveste unike tallet i listen er 8.

Input	Output	Kommentarer
6 1 2 3 3 2 1	-1	Ingen har skrevet ned et unikt tall, så her er det ingen vinner.

## Warpzone

Videospillet Super Francesco er det mest populære spillet å streame for tiden. Du er ganske flink til å spille det, men for å få flest mulige seere så må du være blandt de aller raskeste til å fullføre spillet, og det er ikke så lett!

Spillet har  $N$  baner nummerert fra 1 til  $N$ . Når man har fullført bane  $k$  ( $1 \leq k \leq N - 1$ ) så går man videre til bane  $k + 1$ . Fullfører man bane  $N$  er naturligvis spillet ferdig.



Hver bane, untatt den siste, har også en *warpzone* helt ved slutten, som gir deg 3 snarveier til starten av andre baner (eller sender deg tilbake til banen du var på). Ved å bruke disse på en effektiv måte kan du ofte fullføre spillet mye raskere enn ved å spille gjennom alle  $N$  banene.

Du lurer på hvor mange baner du er nødt til å spille gjennom for å fullføre spillet.

## Input

Første linje inneholder et heltalltall  $N$  - antall baner i spillet.

Deretter følger  $N - 1$  linjer. Den  $i$ 'te av disse inneholder tre tall  $A_i$ ,  $B_i$  og  $C_i$  ( $1 \leq A_i, B_i, C_i \leq N$ ) - de tre banene du kan komme til ved å bruke warpzonen på slutten av bane  $i$ .

## Output

Skriv ut et tall  $b$  - det minste antall baner du er nødt til å spille for å fullføre Super Francesco.

## Begrensninger

$$2 \leq N \leq 100,000$$

**Tidsbegrensning:** 2 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	31	$N \leq 75$
Gruppe 2	33	Ingen snarveier går bakover ( $A_i, B_i, C_i \geq i$ for alle $i$ )
Gruppe 3	36	Ingen andre begrensninger

## Eksempler

Input	Output	Kommentarer
4 1 1 1 2 4 2 3 3 4	3	Alle snarveiene i bane 1 tar deg bare tilbake til bane 1, så disse er det ikke noe poeng i å bruke. Du bør derfor bare fortsette videre til bane 2. På slutten av bane 2 er det en snarvei som lar deg hoppe over bane 3 og gå rett til bane 4. Du bør bruke denne. Når du har fullført bane 4 er spillet ferdig.

Input	Output	Kommentarer
10 7 6 5 1 5 6 2 1 2 10 8 1 2 3 2 2 6 4 3 2 2 1 3 9 2 4 5	4	Det raskeste er å fullføre banene 1, 6, 4 og 10.

## Kuponger

Som vanlig skal elevrådet arrangere en påskebrunsj, og du har fått i oppgave å handle inn maten. Dessverre er elevråds-kassa helt tom men du slipper å gå tomhendt til butikken alikevel! Økonomiansvarlig i elevrådet har nemlig skaffet deg rabattkuponger til matbutikken!

Hver kupong er på formen “ $r$  kr rabatt på handlevogna så lenge sluttsummen blir  $m$  kr eller mer”. Sluttsummen betyr prisen du handler for minus alle rabattene du bruker. F.eks. kan du ikke bruke en “10 kr rabatt på handlevogna så lenge sluttsummen blir 100 kr eller mer”-kupong hvis du handler varer for nøyaktig 100 kr, da kupongen i dette tilfellet ville redusert sluttsummen til 90 og dermed ugyldiggjøre seg selv. Du må faktisk handle varer for minst 110 kr for å få brukt denne kupongen.

Butikken lar deg bruke så mange kuponger du vil, men hver kupong kan maks brukes én gang. Rekkefølgen på kupongene du bruker har ingenting å si. Du har fått oppgitt nøyaktig hva du skal kjøpe, og får ikke lov til å legge til flere varer, selv hvis det potensielt hadde gjort handleturen billigere.

Gitt prisen på handlevoggen før rabatter, finn det laveste beløpet du trenger å betale.



4 x 0,21	0,84
Sosetalltetjes	1,99
Mandarinen	1,99
Sukkeravfels	1,25
Boeren leverworst	0,99
Bruine bollen, 10x50g	1,99
IJsteet	1,99
Cola 6x0,5l	1,92
Kipkluffjes	2,14
Cherrytomaten	0,89
Bøtehamvorst rond	0,89
Achterham 'BL'	1,19
Schouderham rond	0,99
Vanilleyoghurt hv 3 x 0,92	2,76
Vanillevilt	0,81
Bananen	1,48

## Input

Første linje inneholder to heltall  $N$  og  $K$ , prisen på handlevoggen og antall kuponger du har. Deretter følger  $K$  linjer med to heltall.  $r_i$  og  $m_i$  - henholdsvis rabatten og minimumskravet på kupong nummer  $i$ .

## Output

Ett heltall - laveste beløp du må betale.

## Begrensninger

$$1 \leq N \leq 10\,000$$

$$1 \leq K \leq 50$$

$$0 \leq r_i, m_i \leq 10\,000 \text{ for alle } 1 \leq i \leq K$$

Tidsbegrensning: 1 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	10	Alle kupongene er like ( $r_i = r_j$ og $m_i = m_j$ for alle $1 \leq i, j \leq K$ )
Gruppe 2	12	Alle kupongene har samme verdi ( $r_i = r_j$ for alle $1 \leq i, j \leq K$ )
Gruppe 3	12	Alle kupongene har samme minimumskrav ( $m_i = m_j$ for alle $1 \leq i, j \leq K$ )
Gruppe 4	30	$K \leq 20$
Gruppe 5	36	Ingen andre begrensninger

## Eksempler

Input	Output
60 3 20 30 20 30 20 30	40

Input	Output
100 4 7 100 10 50 15 75 20 76	75

## Regnskapssvindel

Det er ikke lett å drive en oppstartsbedrift. Noen investorer krever at selskapet skal være lønnsomt helt fra starten av - noe som naturligvis er ganske vanskelig. Heldigvis har investorene sjeldent tid til å sjekke alle detaljene, så hvis du bare “glemmer” bort noen utlegg når du presenterer regnskapet for de, så vil de som regel ikke merke det.



Du har en liste med inntekter og utgifter som bedriften din har hatt, i kronologisk rekkefølge, med inntekter representert som positive tall og utgifter om negative tall.

Du ønsker å fjerne så få tall som mulige fra denne listen slik at det ser ut som at bedriften aldri har vært i minus. Med andre ord så skal alle mulige del-lister som begynner på starten av listen og går til et vilkårlig punkt ha en totalsum som er minst lik 0.

Hvor mange tall er du nødt til å fjerne?

### Input

Første linje inneholder 1 heltalltall  $N$ : antall transaksjoner (inntekter og utgifter) som bedriften din har i regnskapet.

Deretter følger  $N$  linjer som hver beskriver en transaksjon - et positivt tall for en inntekt og negativt tall for en utgift.

### Output

Et tall  $K$ : antall elementer som må fjernes fra regnskapet for at det vil tilfredsstillere investorene.

### Begrensninger

$$1 \leq N \leq 100,000$$

Alle tallene  $x$  i regnskapet tilfredsstiller  $-1\,000\,000 \leq x \leq 1\,000\,000$

**Tidsbegrensning** 1 s.

Merk at summen av talene kan være for store til å få plass til i et 32-bits heltall. Det kan dermed være lurt å benytte seg av 64-bits heltall (`long long` i C++, eller `long` i Java. Ikke noe spesielt trengs å gjøre i Python.)

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	20	$N \leq 20$
Gruppe 2	36	$N \leq 1\,000$
Gruppe 3	44	Ingen andre begrensninger

## Eksempler

Input	Output
9 -1 5 -4 -2 2 -5 -3 10 -9	3

**Forklaring:** Dersom du fjerner bokføringene på -1, -4 og -5, så sitter du igjen med

Transaksjon	Balanse på konto etter transaksjon
5	5
-2	3
2	5
-3	2
10	12
-9	3

og er dermed aldri i minus.



## Myntgraf

Anders har et gigantisk rutenett og  $N$  brikker nummerert fra 1 til  $N$ . Han fordeler brikkene utover rutenettet slik at hver brikke ligger i en av rutene og at ingen av rutene inneholder mer enn 1 brikke. Deretter noterer han seg alle par av brikker som ligger på enten samme rad eller samme kolonne. F.eks for utsnittet under så vil han notere parene (1 4), (1 5), (4 5), (1 6), (4, 2), (5, 3), (6, 3). (Han kan evt. ha notert parene i en annen rekkefølge eller bytte om på hvilket tall han skriver først i noen av parene, men han vil alltid ha notert alle disse parene).

1	4		5
6			3
	2		

Gitt en slik beskrivelse, gjenskap en plassering av myntene som har denne beskrivelsen. Dersom det ikke er mulig å plassere myntene slik at beskrivelsen er gyldig, skriv ut UMULIG.

### Input

Første linje inneholder  $N$  - antall mynter, og  $M$  - antall par med mynter på listen.

Deretter følger  $M$  linjer med to tall  $A_i$  og  $B_i$  ( $1 \leq A_i, B_i \leq N$ ) - nummerene på to mynter som ligger enten på samme rad eller på samme kolonne.

### Output

Dersom det ikke finnes noen gyldig plassering av myntene skriv ut ordet UMULIG.

Hvis det finnes gyldige plasseringer, så skriv ut en vilkårlig gyldig plassering av myntene. En slik beskrivelse består av  $N$  linjer. Den  $i$ -te av disse skal inneholde et tallpar  $r_i$  og  $k_i$  skilt med mellomrom - henholdsvis raden og kolonnen til brikke nummer  $i$ . Alle koordinatene du bruker må ha  $0 \leq r_i, k_i \leq 1\,000\,000\,000$ .

## Begrensninger

$$1 \leq N \leq 20\,000$$

$$1 \leq M \leq 40\,000$$

**Tidsbegrensning:** 2 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	8	$N \leq 5$
Gruppe 2	15	Ingen brikker er med i flere enn 2 par
Gruppe 3	32	$N \leq 20, M \leq 20$
Gruppe 4	20	$N \leq 1\,000, M \leq 1\,000$
Gruppe 5	25	Ingen andre begrensninger

## Eksempler

Input	Output	Kommentarer
6 7 1 4 1 5 4 5 1 6 4 2 5 3 6 3	0 0 3 1 2 3 0 1 0 3 2 0	Dette er eksempelet i tegingen. Merk at det finnes andre gyldige plasseringer av myntene.

Input	Output
4 4 1 2 1 3 2 3 2 4	10 10 15 10 20 10 15 20

Input	Output	Kommentarer
4 3 1 2 1 3 1 4	UMULIG	Mynt 1 kan ikke dele rad eller kolonne med alle de tre andre myntene uten at noen av de også deler rad eller kolonne med hverandre.

Input	Output
4 5 1 2 1 3 2 3 2 4 3 4	UMULIG