Rektangelrot

Roar, rekkverksansvarlig i Ringebu har fått problemer med rekkverksplanene sine. Han skulle motta 2 hjørnekoordinater for rektangulære områder som skulle få rekkverk, men han har i stedet mottatt koordinater for to rektangler som overlapper!

Rekkverksansvarlig Roar ønsker å vite hvor mye rekkverk han trenger, for å legge rekkverk rundt området som dekkes av rektanglene til sammen, slik at Ringebus rektangulære rekkverksbyrå kan føre hvor mange meter rekkverk han bruker.

Input

Første linje inneholder et enkelt heltall N - antall områder Roar har mottatt koordinater for.

Deretter følger beskrivelsene til de N områdene Roar har mottat koordinater for. Hvert område består av to rektangler, og et rektangel beskrives av et par av motsatte hjørnepunkter. Beskrivelsen til område nummer i består derfor av to linjer, som hver består av fire heltall x_1, y_1, x_2, y_2, som bestemmer to hjørnepunkter (x_1, y_1) og (x_2, y_2).

Her er et forslag til hvordan man kan lese inn input på denne oppgaven i Python:

```python
N = int(input())

for i in range(N):
    r1x1, r1y1, r1x2, r1y2 = map(int, input().split())
    r2x1, r2y1, r2x2, r2y2 = map(int, input().split())
```

Output

N linjer, hver med et enkelt heltall - hvor mye rekkverk Roar må bruke for å legge rekkverk rundt område i.

Begrensninger

$1 \leq N \leq 1000$

$0 \leq x_1, y_1, x_2, y_2 < 10\ 000$

$x_1 < x_2$

$y_1 < y_2$

Tidsbegrensning. 1 s

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 1	25	Rektanglene overlapper fullstendig
Gruppe 2	15	De to rektanglene har alltid samme x_1-koordinat og samme x_2-koordinat
Gruppe 3	15	De to rektanglene har alltid samme y_1-koordinat og samme y_2-koordinat
Gruppe 4	45	Ingen andre begrensninger

Løsning for testsettgruppe 1

Denne oppgaven kommer med en komplett løsning for testsettgruppe 1 slik at det skal være lettere å komme igang med runde 2 og bli kjent med konkurransesystemet.

I testsettgruppe 1 er begrensningen at rektanglene overlapper fullstendig. Dvs. at alle koordinatene til det ene rektangelet alltid er innenfor det andre rektangelet. Da kan vi løse oppgaven ved å regne ut omkretsen til de to rektanglene seperat og ta den som er størst.

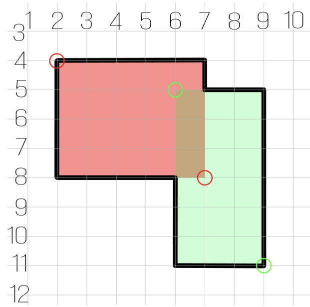
Det kan gjøres i Python slik som dette:

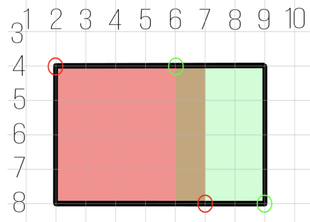
```
N = int(input())

for i in range(N):
    r1x1, r1y1, r1x2, r1y2 = map(int, input().split())
    r2x1, r2y1, r2x2, r2y2 = map(int, input().split())

    o1 = 2 * abs(r1x1 - r1x2) + 2 * abs(r1y1 - r1y2)
    o2 = 2 * abs(r2x1 - r2x2) + 2 * abs(r2y1 - r2y2)

    print(max(o1, o2))
```

Hint for testsettgruppe 2 og 3

I testsettgruppe 2 og 3 overlapper rektanglene henholdsvis kun i y- og x-retningen. Se det andre eksemplet. Da betyr det at omkretsen i x- og y-retningen for hele området er det samme som for bare en av rektanglene og vi må kun bekymre oss for den andre retningen. Der må vi bare se på hvor mye de to rektanglene overlapper.

Eksempler

Input	**Output**	**Kommentarer**
1 2 4 7 8 6 5 9 11	28	

Input	**Output**	**Kommentarer**
1 2 4 7 8 6 4 9 8	22	

Input	**Output**
3 10 10 20 20 15 15 35 35 5 7 35 17 1 3 6 20 3 7 4 8 1 2 11 13	100 102 42

Input	**Output**	**Kommentarer**
2 1 1 10 10 4 4 5 5 1 3 7 8 3 3 5 7	36 22	Dette eksemplet beskriver input som er gyldig for testsettgruppe 1 og 4

Minotaurens nye ide

Minotauren begynner å bli lei av å vokte labyrinten, og ser etter en ny tvist. Planen er å lage et gameshow der deltakeren og minotauren blir plassert forskjellige steder i labyrinten, for så å la minotauren jage deltakeren. Labyrinten består av N kryss, og M ganger. Hver gang kobler et kryss til et annet kryss. Det er alltid mulig å gå fra et hvilket som helst kryss til alle andre via en rekke med ganger. Ingen kryss har en gang fra seg selv som går tilbake til det samme krysset. Mellom alle par med kryss finnes det maksimalt én gang.

Deltakeren får velge selv i hvilket kryss de vil starte. Deretter får minotauren velge sitt eget startkryss, blant alle andre kryss. Minotauren og deltakeren løper like fort, og de vet begge hvor hverandre er til enhver tid.

Spillet er rigget, så det er ingen vei ut av labyrinten. Likevel kan det hende at minotauren ender opp med å jage deltakeren i en ring for alltid. Sånt har ikke minotauren tid til. Minotauren ønsker derfor å lage en liste over kryss der deltakeren kan starte, hvor minotauren kan hindre deltakeren fra å hale ut spillet til evig tid.

Input

Første linje inneholder tallene N og M, antall kryss og antall ganger i labyrinten. De neste M linjene inneholder to tall i og j, som betyr at det er en gang mellom den i-ende og j-ende krysset i labyrinten.

Output

En liste med alle kryss minotauren kan tillate at deltakeren kan starter i, slik at minotauren kan garantere at spillet tar slutt. En linje per kryss. Kryssene må være i stigende rekkefølge.

Begrensninger

$2 \leq N \leq 100\ 000$

$1 \leq M \leq 100\ 000$

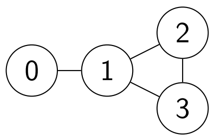
$0 \leq i, j < N$

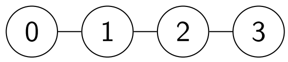
Tidsbegrensning. 2 s

Testsettgruppe	**Poeng**	**Ytterlige begresninger**
Gruppe 1	10	Hvert kryss i labyrinten er koblet til maksimalt to andre kryss
Gruppe 2	10	$M = N - 1$

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 3	10	Hvert kryss i labyrinten er koblet til alle de andre kryssene
Gruppe 4	10	Hvert kryss i labyrinten er koblet til minst 2 andre kryss
Gruppe 5	20	Alle utenom K kryss er koblet sammen på samme måte som i testsett 4. De resterende K kryssene er koblet til nøyaktig ett annet kryss
Gruppe 6	40	Ingen andre begrensninger

Eksempler

Input	Output	Kommentarer
4 4 0 1 1 2 1 3 2 3	0	Labyrinten ser ut som følger:  Dette er et gyldig input i testsett 5 og 6.

Input	Output	Kommentarer
4 3 0 1 1 2 2 3	0 1 2 3	Labyrinten ser ut som følger:  Dette er et gyldig input i testsett 1, 2 og 6.

Rettferdig Fordeling

Arne og Berit spiller et spill der de skal fordele en liste med N tall mellom seg etter tur. Tallene skal deles ut fra toppen av listen og nedover, og de bytter på å bestemme hvem som skal få neste tall.

Når det er Arne sin tur, kan han enten velge å ta øverste tall til seg selv, eller gi tallet til Berit. Det fortsetter å være Arne sin tur helt til han har gitt M tall til seg selv. Da blir det Berit sin tur til å velge hvem som skal få neste tall, helt til hun har gitt M tall til seg selv, og det blir Arnes tur igjen. Slik fortsetter de nedover listen, helt til alle tallene er delt ut.

Målet med spillet er å få høyest mulig sum når hele listen er delt ut. Gitt listen de skal spille med, og at Arne begynner, hvilken poengsum vil de to ende opp med dersom begge to spiller helt perfekt?

Input

En linje med to heltall N og M - lengden på listen og antall tall en spiller gir til seg selv hver tur. Deretter følger N linjer med ett heltall L_i hver - tall nummer i på listen.

Output

En linje med to heltall A og B - poengsummene Arne og Berit sitter igjen med dersom de spiller helt perfekt begge to.

NB: Svarene kan bli større enn 2^{31}, så du må bruke `long long` i C++, og `long` i Java. I Python vil vanlig `int` fungere fint.

Begrensninger

$1 \leq N \leq 100\ 000$

$1 \leq M \leq 20$

$-1\ 000\ 000 \leq L_i \leq 1\ 000\ 000$

Tidsbegrensning. 3 s

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 1	15	$M = 1$, Listen består av positive tall i synkende rekkefølge
Gruppe 2	15	$M = 1 \quad N \leq 10$
Gruppe 3	30	$M = 1$, Listen består kun av positive tall

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 4	20	$N \leq 1000$
Gruppe 5	20	Ingen andre begrensninger

Eksempler

Input	Output	Kommentarer

`6 1`
`1`
`1`
`5`
`4`
`3`
`6`
	`11 9`	Tur 1: Arne gi to 1ere til Berit, for å kunne ta 5 selv.
		
		Tur 2: Berit tar 4, fremfor å gi bort 4 og 3 for å få 6.
		
		Tur 3: Arne får valget mellom 3 og 6, og gir 3 til Berit, og 6 til seg selv.

Input	Output	Kommentarer

`8 2`
`60`
`50`
`70`
`50`
`20`
`4`
`7`
`-10`
	`137 114`	Tur 1: Arne tar 60, gir 50 til Berit, og tar de neste 70.
		
		Tur 2: Berit tar 50 og 20
		
		Tur 3: Arne gir 4 til Berit, tar 7 og gir -10 til Berit

Tognett

I landet Vidbredistan ligger det N byer, spredd over et stort landområde. Mellom byene går det M flyruter, der hver rute går frem og tilbake mellom nøyaktig to byer. Det finnes maks én flyrute mellom det samme paret av byer.

I et forsøk på å redusere flytrafikk, har myndighetene bestemt seg for å bygge hurtigtog. Det skal bygges K togforbindelser, der hver forbindelse går mellom to byer.

Togforbindelsene bygges én av gangen, og hver gang det blir mulig å komme seg mellom to byer med kun tog, blir det forbudt å fly mellom de to byene og en eventuell direkte flyforbindelse mellom de to byene blir nedlagt. For hver nye togforbindelse som bygges, ønsker myndighetene å vite hvor mange flere flyruter som tvinges til å legge ned. Det bygges aldri mer enn en togforbindelse mellom det samme paret av byer.

Input

Første linje inneholder tre heltall N, M og K - antall byer i Vibredistan, antall flyruter og antall nye togforbindelser som skal bygges.

Deretter følger M linjer. Hver med to heltall $0 \leq a_i < N$ og $0 \leq b_i < N$ - flyrute nummer i.

Sist følger det K linjer. Hver med to heltall $0 \leq u_j < N$ og $0 \leq v_j < N$ - togforbindelse nummer j.

Output

K linjer, hver med et enkelt heltall - antall nye flyruter som må legges ned når togforbindelse j blir bygget.

Begrensninger

$2 \leq N \leq 200\ 000$

$1 \leq M \leq 200\ 000$

$1 \leq K \leq 100\ 000$

Tidsbegrensning. 2 s

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 1	15	$N \leq 50, K \leq 50$
Gruppe 2	20	$M \leq 2000, K \leq 2000$

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 3	30	$N \leq 400$ og det er nøyaktig én flyrute mellom hvert par av byer
Gruppe 4	35	Ingen andre begrensninger

Eksempler

Input	Output
4 4 2 0 1 1 2 2 3 3 1 3 0 1 3	0 2

Basehopp

I klippefjellene er basehopp en populær aktivitet. Hoppområdet deres er en strak linje, og hopperne liker å hoppe så langt langs denne linjen som mulig. Nå planlegger hoppforeningen å bygge tårn for å kunne hoppe enda lenger, men de er usikre på hvor det lønner seg å bygge.

Foreningen har gitt deg høydekurven langs hoppområdet, med terrengets høyde for hver meter i luftlinje. I luften kan en basehopper gå én meter bortover, for hver meter de faller, helt til de lander trygt på bakken. En hopper kan ikke forlate hoppområdet. Se figuren til eksempel 1.

Input

To heltall N og M – lengden på hoppområdet i meter, og antall forslag til tårn.

Én line med N heltall y_i – terrengets høyde for hver meter i $(0 \leq i < N)$.

M liner med to heltall x_j og h_j – posisjonen og høyden til hvert tårn j $(0 \leq j < M)$.

Output

M linjer, én for hvert tårnforslag j, med ett heltall d_j på hver linje – den maksimale horisontale avstanden en hopper kan nå fra toppen av tårn j.

Begrensninger

$1 \leq N \leq 200000$

$1 \leq M \leq 100000$

$0 \leq y_i, h_j \leq 10^9$

$0 \leq x_j < N$

Tidsbegrensning. 1 s

Alle tårn bygges over bakkenivå $(h_j > y_{x_j})$.

Testsettgruppe	Poeng	Ytterlige begresninger
Gruppe 1	10	$N, M \leq 1000$
Gruppe 2	20	Toppen av alle tårn er maks 100 meter over havet
Gruppe 3	30	Alle tårn bygges i $x_j = 0$
Gruppe 4	40	Ingen andre begrensninger

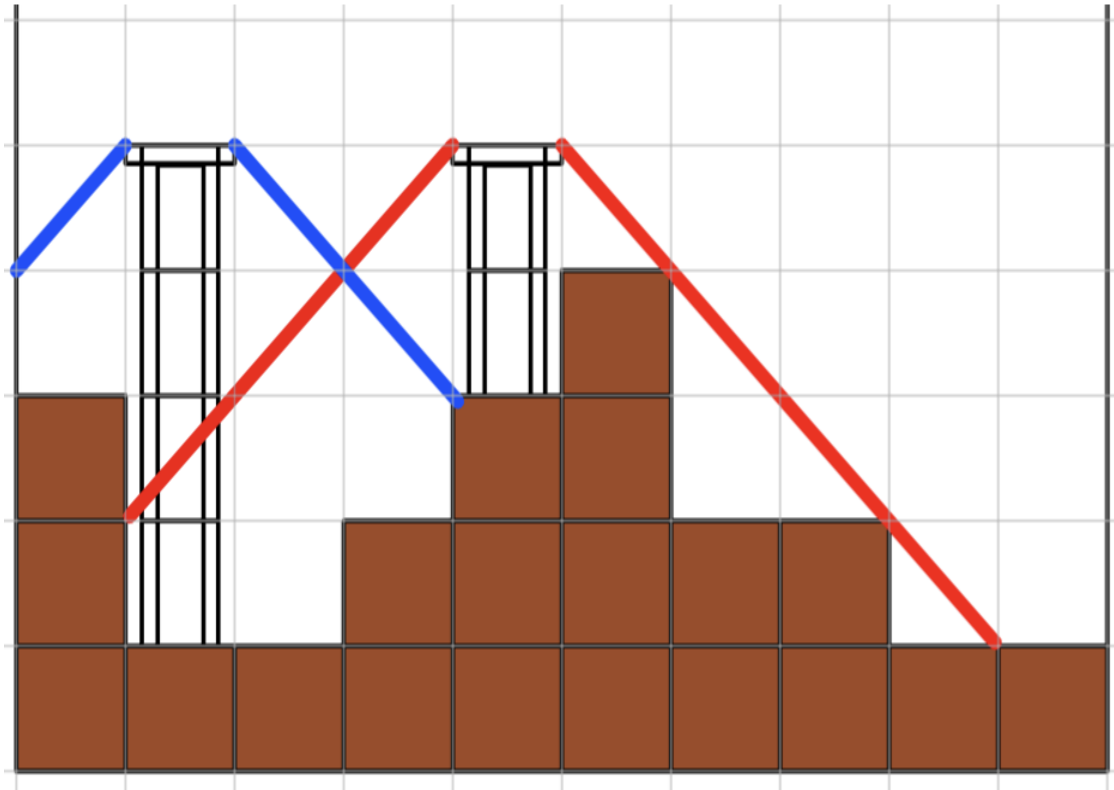


Figure 1: illustrasjon av eksempelinput

Eksempler

Input	Output	Kommentarer
10 2 3 1 1 2 3 4 2 2 1 1 1 5 4 5	2 4	Fra det første tårnet kan man hoppe 1 meter til venstre, før hopperen når kanten av hoppeområdet, eller 2 meter til høyre. Fra det andre tårnet kan hopperen hoppe 3 meter mot venstre, eller 4 meter til høyre.