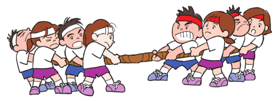
Tautrekking



En tautrekningskonkurranse foregår ved at man har 2 lag hver med K deltagere og et tau som er $2K$ meter langt. Deltagerene på hvert lag står på hver sin side av tauet og prøver å trekke tauet vekk fra det andre laget.

For at konkurransen skal være mest mulig visuelt tilfredsstillende så bør den ene halvdelen av tauet være rødt og den andre halvdelen blått. Dermed blir det lett for alle å se hvilken del av tauet som tilhører hvilket lag.

Du har et tau som er N meter langt og hvor hver meter er farget enten rødt eller blått. Du ønsker å klippe ut en del av dette tauet slik at du får et tau som du kan bruke i en størst mulig tautrekkekonkurranse. Men andre ord skal du finne største mulige verdi K slik at det finnes en del av tauet som er $2K$ meter langt og hvor den ene halvdelen er rød og den andre halvdelen er blå.

Input

Første linje inneholder ett heltalltall N lengden på det opprinnelige tauet. Andre linje inneholder en streng bestående av nøyaktig N tegn som beskriver tauet. Denne strengen inneholder bokstaven *r* hvis tauet er rødt på denne posisjonen, eller *b* hvis tauet er blått på denne posisjonen.

Output

Skriv ut et tall K - det størte antall personer du kan ha på hvert lag i tautrekkekonkurransen

Begrensninger

$2 \leq N \leq 100\ 000$

Tauet vil alltid ha minst én rød seksjon og minst én blå seksjon.

Tidsbegrensning: 1 s.

Testsettgruppe	**Poeng**	**Ytligere begrensninger**
Gruppe 1	45	$N \leq 50$
Gruppe 2	20	$N \leq 500$
Gruppe 3	35	Ingen andre begrensninger

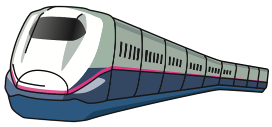
Eksempler

Input	**Output**	**Kommentarer**
`6` `bbbrrr`	`3`	Her kan du bruke hele det opprinnelige tauet

Input	**Output**	**Kommentarer**
`10` `brrrbbrbbb`	`2`	Den beste taubiten er rrbb

Input	**Output**
`20` `rbbbbbbbbrbrrrrrrrrb`	`1`

Togtur



Kjetil har vært på Python-seminar og brukt alt for mye penger på lærebøker. Han må dermed finne billigste måte å reise hjem igjen med tog.

Tognettet er består av N byer som er forbundet med K togforbindelser. Hver forbindelse knytter sammen 2 byer, har en tilhørende billetpris, og kan brukes i begge retninger.

Kjetil har også V gratisbilletter som han kan bruke på en hvilken som helst strekning for å slippe å betale for biletten. Hva er billigste måte å komme hjem på?

Input

Første linje inneholder tre heltalltall N, M og V, henholdsvis antall byer, antall togforbindelser og antall gratisbiletter som Kjetil har. Deretter følger M linjer som hver beskriver en togforbindelse. Disse linjene inneholder 3 heltall a_i, b_i og k_i, henholdsvis de to byene som forbindelsen går mellom og billetprisen for denne strekningen.

Byene er nummerert fra 0 til og med $N - 1$. Python-seminaret har vært i by 0 og Kjetil bor i by 1.

Output

Skriv ut et tall S - den minste totale prisen du må ut med for å komme deg hjem.

Begrensninger

$2 \leq N \leq 2\,500$

$1 \leq M \leq 10\,000$

$0 \leq V \leq 50$

$0 \leq a_i, b_i < N$ for alle i

$1 \leq k_i \leq 100\,000$ for alle i

Det vil alltid være mulig for Kjetil å komme seg hjem.

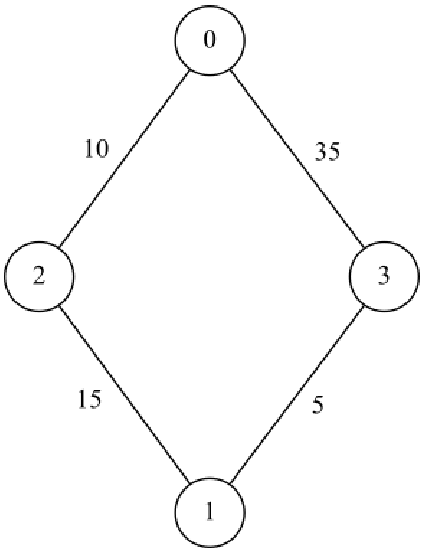
Tidsbegrensning: 2 s.

Testsettgruppe	**Poeng**	**Ytligere begrensninger**
Gruppe 1	20	$V = 0$
Gruppe 2	13	$V = 1; N \leq 500; M \leq 1\ 000$
Gruppe 3	17	$V = 1$
Gruppe 4	20	$N \leq 500; M \leq 1\ 000$
Gruppe 5	30	Ingen andre begrensninger

Eksempler

Input	**Output**	**Kommentarer**

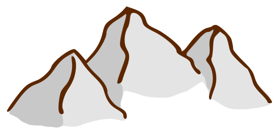
`4 4 0`
`2 0 10`
`0 3 35`
`1 2 15`
`1 3 5`
	`25`	Se diagrammet under over hvordan tognettet ser ut. Den billigste måten å komme fra 0 til 1 på er ved å betale $10 + 15 = 25$.



Input	**Output**	**Kommentarer**

`4 4 1`
`2 0 10`
`0 3 35`
`1 2 15`
`1 3 5`
	`5`	Her ser tognettet likt ut, men Kjetil har en gratisbillet han kan bruke.

Snøballtopografi



Du er i helikopter over skifjellet klar til å hoppe ut med snøbrettet ditt. Problemet er bare at bakken under deg er helt hvit, og dermed klarer du ikke å se hvor terrenget er høyest og lavest. Helikopterets GPS viser titusendels lengdegrad og breddegrad, så du kan modellere skifjellet som et rutenett med $M \times N$ ruter, der M er bredde og N er høyde. Nord er negativ y-retning på rutenettets koordinatsystem, sør er positiv y-retning, vest er negativ x-retning og øst er positiv x-retning. For å modellere høyde forenkler vi og antar at hver rute har én høyde målt i meter over havet.

Fra helikopteret slipper du K snøballer, noterer hvor de lander, og hvordan de ruller. Du følger alle snøballene helt til de stopper, og noterer hvilke ruter de besøker. En snøball kan kun rulle fra en rute til en naborute dersom naboruten er minst én meter lavere. Dersom det er flere slike naboruter vil snøballen rulle tilfeldig til en av dem. Dersom det ikke er noen slike naboruter stopper snøballen.

Du har lyst til å stå på ski så mange høydemeter som mulig.

Gitt hvor de K snøballene lander, og rutene de besøker til de stopper, finn to koordinater i rutenettet med størst mulig garantert høydeforskjell.

Input

Første linje inneholder tre heltall M, N og K - henholdsvis bredden på rutenettet, høyden på rutenettet, og antall snøballer som er sluppet.

Deretter følger $2 \times K$ linjer. Hvert par av disse linjene beskriver en av snøballene.

Den første linjen i et slikt par inneholder tre heltall X_i, Y_i og L_i - Snøballen sin start-posisjon og lengden på rullingen. Den andre linjen i paret er en streng på L_i tegn, hvor hvert tegn er en himmelretning: `N` (nord), `S` (sør), `W` (vest) eller `E` (øst) som beskriver stien som snøballen ruller fra rute til rute fram til den stopper.

Output

Skriv ut en linje med 5 tall: D X_h Y_h X_l Y_l

D skal være den største tallet slik at man vet at høydeforkjellen mellom den øverste og nederste er på minst D meter.

(X_h, Y_h) og (X_l, Y_l) skal være koordinatene til to ruter, hvor man vet at rute (X_h, Y_h) er minst D meter høyere enn rute (X_l, Y_l)

Dersom flere par av koordinater har samme nedre grense for høydeforskjell blir alle godkjent.

Begrensninger

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

$M \times N \leq 2\ 000\ 000$

$1 \leq K \leq 10\ 000$

$0 \leq X_i < M$ for alle i

$0 \leq Y_i < N$ for alle i

$1 \leq L_i \leq 10\ 000$ for alle i

Summen av alle L_i er mindre enn eller lik 200 000

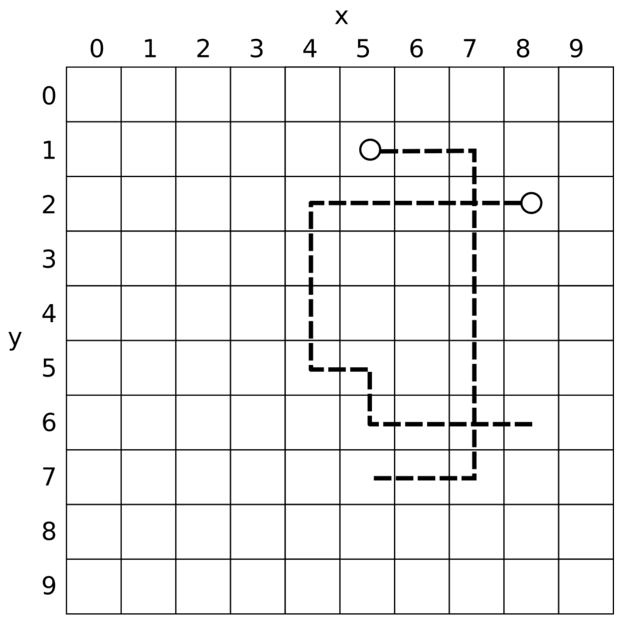
Ingen snøballer vil rulle ut av rutenettet.

Tidsbegrensning: 1 s.

Testsettgruppe	**Poeng**	**Ytligere begrensninger**
Gruppe 1	19	$M, N \leq 10$; $K \leq 10$; $\text{sum}(L_i) \leq 30$
Gruppe 2	19	$K = 2$
Gruppe 3	11	Hver rute har maksimalt én naborute som er 1 eller flere meter lavere enn seg.
Gruppe 4	51	Ingen andre begrensninger

Eksempler

Input	Output	Kommentarer
10 10 2 8 2 12 WWWWSSSESEEE 5 1 10 EESSSSSSWW	16 5 1 5 7	Tegningen under viser nedslagsfeltet til de to snøballene som sirkler, og stiene de ruller før de stopper som stripla linjer.



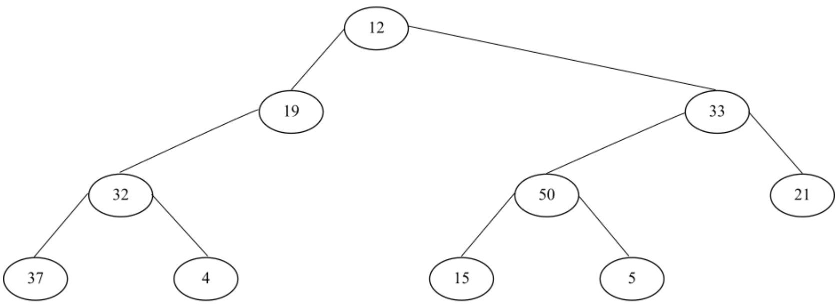
Glatte trær

Et binært tre er en datastruktur bestående av *noder*. Hver node kan ha opptil to *barn* - ett venstre og ett høyre - som også er noder. En node kan altså ha enten ingen barn, kun et venstre barn, kun et høyre barn, eller både et venstre og høyre barn. En av nodene kalles *roten*, og er den eneste noden som ikke er barnet til noen av de andre.

I denne oppgaven ser vi på binære trær hvor hver node har et tilhørende tall. Vi kan skrive ut alle tallene i det som kalles *infix rekkefølge* ved å kalle følgende rekursive metode (skrevet i pseudokode) med rot-noden som parameter:

```
infix_utskrift(node):
    hvis node har et venstre barn:
        infix_utskrift(node.venstre_barn)
    print(node.verdi)
    hvis node har et høyre barn:
        infix_utskrift(node.høyre_barn)
```

For eksempel viser diagrammet under et binært tre, og den tilhørende infix rekkefølgen av dets tall.



37	32	4	19	12	15	50	5	33	21

Vi definerer ruheten til å være summen av absoluttverdiene av alle differanser mellom påfølgende tall i infix rekkefølgen dets. Ruheten til dette treet er dermed

$$|37-32|+|32-4|+|4-19|+|19-12|+|12-15|+|15-50|+|50-5|+|5-33|+|33-21| =$$
$$5+28+15+7+3+35+45+28+12 = 178$$

Ruheten til et tre kan endres hvis man bytter om på hvilken side barna til en node befinner seg på. For eksempel, ved å bytte om på de to barna til noden med verdi 32 så får treet i stedet en ruhet på 181. For noder som bare har ett barn vil vi også tillate at man endrer hvilken side dette barnet befinner seg på. F.eks. kan du flytte noden med verdi 32 til å være høyre barn av node 19. Slike endringer kan også endre ruheten til treet.

Hva er den laveste ruheten som det er mulig å oppnå for et tre ved å kun gjøre disse type endringene?

Input

Første linje inneholder ett heltalltall N antall noder i treet. Deretter følger N linjer, hver som beskriver en node i treet. Den i-te av disse beskriver node nummer i (linjer og noder nummerert fra 0 til $N-1$). En slik linje består av tre heltall $verdi_i$, $venstre_i$, $høyre_i$. $verdi_i$ er verdien til node i, mens $venstre_i$ og $høyre_i$ er nummeret (altså ikke verdien), til nodene som er henholdvis venstre og høyre barn av node nummer i. Dersom node i ikke har et barn til venstre (høyre), så vil $venstre_i$ ($høyre_i$) være tallet -1.

Node 0 vil alltid være roten i treet.

Output

Skriv ut et tall - den minste oppnåelige ruheten til treet, ved å gjøre en sekvens av slike tillatte endringer.

Begrensninger

$1 \leq N \leq 1\ 000$

$0 \leq verdi_i \leq 1\ 000\ 000$ for alle i.

Tidsbegrensning: 2 s.

Testsettgruppe	**Poeng**	**Ytligere begrensninger**
Gruppe 1	10	$N \leq 16$; $verdi_i \leq 100$ for alle i
Gruppe 2	20	$N \leq 100$; $verdi_i \leq 100$ for alle i
Gruppe 3	30	$N \leq 100$
Gruppe 4	40	Ingen andre begrensninger

Eksempler

Input	Output
5 9 -1 1 6 -1 4 2 -1 -1 10 -1 -1 1 2 3	15

Input	Output	Kommentarer
10 12 1 2 19 3 -1 33 4 5 32 6 7 50 8 9 21 -1 -1 37 -1 -1 4 -1 -1 15 -1 -1 5 -1 -1	172	Dette er treet vist i diagrammet

Slange

Slangen Sivert skal gå i dvale for vinteren, og lurer på hva det mest komfortable stedet den kan legge seg i tunnelen sin er. Tunnellen har lengde N og Sivert har lengde L. Det betyr at Sivert må finne L etterfølgende ruter i tunellen sin hvor den skal sove. Hver bit av tunellen har en ruhet uttrykt ved et ikke-negativt heltall. Den totale ukomforten til Sivert er summen av ruhetene i alle rutene han ligger på. Sivert har K puter som han kan bruke for å gjøre deler av tunellen mer komfortabel. Hver av disse reduserer ruheten til en enkelt rute ned til 0.

Hva er den minste mulige ukomforten som Sivert kan oppnå?

Input

Første linje inneholder tre tall, N, L og K slik beskrevet i oppgaveteksten. Andre linje inneholder N tall - ruheten til de N segmentene som utgjør tunellen i rekkefølge.

Output

Skriv ut et tall den minste ukomforten som det er mulig å oppnå

Begrensninger

$1 \leq N \leq 200\ 000$

$1 \leq L \leq N$

$0 \leq K < L$

Tidsbegrensning: 1 s.

Testsettgruppe	**Poeng**	**Ytligere begrensninger**
Gruppe 1	45	$N \leq 50$
Gruppe 2	20	$N \leq 500$
Gruppe 3	35	Ingen andre begrensninger

Eksempler

Input	**Output**	**Kommentarer**
`9 3 1` `3 4 1 9 2 4 5 0 5`	`3`	Legg puten på punktet med ruhet 9 og sov på 1, (9), 2