



Kirkeklokker

Bergen domkirke har fått mye klager på at kirkeklokken slår for mye, og vil nå ha hjelp til hvordan de kan slå klokken færre ganger pr. dag, men forsikre seg om at alle rundt dem kan oppfatte hva tiden er hver gang den blir annonsert.

Kirken vil annonsere hvert kvarter fra og med klokken 06:00 til og med klokken 18:00. En annonsering bestemmes unikt av antall slag med kirkeklokken, minst ett slag.

Videre antar kirken at folk kan huske alle slag som har skjedd den siste timen. Det vil si at du kan anta at alle rundt kirken vil kunne huske de tre foregående annonseringene når et nytt kvarter er gått. Sagt mer formelt, på annonsering nr. $i + 4$ vil slagene som ble slått på annonsering nr. i være glemt, men slagene fra annonseringene $i + 1, i + 2, i + 3$ vil være husket.

Gitt disse antagelsene, gi en instruks til kirken om hvordan de skal annonsere tiden, med færrest mulige slag av klokken totalt.

Input

Denne oppgaven har ingen input. Det betyr at det er mulig å finne en løsning for hånd, og sende inn et program som bare printer løsningen.

Output

Output består av én linje med 49 heltall større enn eller lik 1, hvor heltall nummer i representerer antall slag kirken skal slå på kvarter nummer i mellom 06:00 og 18:00.

Scoring

Denne oppgaven har ingen deloppgaver. I stedetfor får du poeng basert hvor mange slag løsningen din har totalt. Du får den høyeste scoren som løsningen kvalifiserer seg til, av følgende tre muligheter.

- Gyldige løsninger får 1 poeng
- Løsninger som bruker x slag, får $150 - x$ poeng
- Teoretisk optimal løsning får 100 poeng

Merk at vi ikke røper hvor mange slag som er optimalt.

Tidsbegrensning: 2 s



Eksempel

Vi kan for eksempel slå klokken en gang for antall kvarter siden 06:00, pluss 1, da vi må slå minst ett slag pr. annonsering.

1 2 3 4 ... 47 48 49

Denne løsningen bruker 1225 slag og gir derfor 1 poeng.

Paidag

I dag er det pi-dagen, og da må man jo følgelig feire det med pai. Siden dette er en stor folkefest er det viktig at det er nok pai til alle. Derfor har du fått ansvaret for å sikre at det er M paier til feiringen. Ettersom det er for mye jobb å lage så mange paier har N folk tilbudt seg å lage én pai hver. Hver av paiene består av et utvalg av K ingredienser nummerert $0 \dots K - 1$. Dvs. at en pai kan inneholde ingrediens 0, 3, 4, mens en annen kan bare inneholde ingrediens 1. Du har fått oppgitt hvilke ingredienser hver av de N folkene trenger for å lage paien sin. For å spare penger lurer du nå på hva som er minste antall typer ingredienser du må kjøpe inn. Hver gang du kjøper en type ingrediens, en K_i , vil du ha ubegrenset tilgang på den ingrediensen.

Input

Første linje består av tre heltall: N , M , og K som er antall folk som tilbyr seg å lage pai, antall paier som trengs og antall ingredienser som kan bli brukt respektivt.

Videre følger det N linjer. Hver linje starter med et heltall a_i som er antall ingredienser for pai nummer i . Så kommer de forskjellige ingrediensene i pai i på samme linje. Hver ingrediens er representert ved et tall mellom 0 og $K - 1$.

Output

Minste antall typer ingredienser som må kjøpes for å kunne lage M paier.

Begrensninger:

$$1 \leq M \leq N \leq 80\,000$$

$$1 \leq K \leq 18$$

$$1 \leq a_i \leq K$$

Tidsbegrensning: 3 s

Testsettgruppe	Poeng	Ytterligere begrensninger
Gruppe 1	19	$K \leq 3$
Gruppe 2	17	$M = N$
Gruppe 3	33	$N \leq 1000, K \leq 10$
Gruppe 4	31	Ingen andre begrensninger



Eksempel 1

Input	Output	Kommentarer
4 2 5 3 1 2 4 2 1 2 2 0 2 2 3 4	3	Det er to mulige kombinasjoner av ingredienser: 0, 1, 2 eller 1, 2, 4.



Skattekartet

Etter et vellykket ran (med altruistiske intensjoner, såklart) har du gravd ned byttet på et flatt, kvadratisk jorde utenfor Bergen et sted.

I planleggingen av ranet har du tegnet opp et $N \times N$ koordinatsystem som representerer jordet. Her representerer punktene $(1, 1)$ et valgt hjørne, og punktene $(N, 1)$, (N, N) og $(1, N)$ representerer de neste tre hjørnene når vi ser på jordet ovenfra og vi går med klokken. Etter at du har gravd ned skatten på det planlagte stedet så ønsker du å markere enkelte koordinater på jordet med noen av gullmyntene du har til overs fra ranet, slik at du senere kan finne igjen skatten.

Flere år senere, hvor du i mellomtiden har holdt en lav profil, returnerer du til jordet for å grave opp skatten. Du husker hverken himmelretningen du markerte koordinatene på jordet eller på hvilket koordinat du gravde ned byttet, så nå håper du at du har markert koordinatene på en fornuftig måte slik at du med så lite undersøkelser som mulig kan hente inn skatten.

Implementer en algoritme for å markere at en skatt skal graves ned på et gitt $N \times N$ koordinatsystem slik at du senere, uten nøyaktig informasjon om hvor skatten var gravd ned og uten informasjon om hvilken himmelretning koordinatene tilsvarer, finne skatten ved å sjekke færrest mulige koordinater for markering.

Merk at programmet ditt vil bli kjørt i flere omganger, for å forsikre at du ikke lokalt kan lagre koordinatene til skatten.

Input

Første linje består enten av teksten `kartlegg` eller `let`, avhengig av om programmet ditt nå vil bli bedt om å markere visse koordinater eller finne skatten. Om første linje er gitt ved `kartlegg` sier vi at vi er i kartleggingsmodus. Om første linje er gitt ved `let` sier vi at vi er i letemodus.

I kartleggingsmodus vil andre linje inneholde tre heltall, N , X og Y , hvor $1 \leq X \leq N$ og $1 \leq Y \leq N$ slik at (X, Y) er koordinatet til skatten, gitt at koordinatet $(1, 1)$ refererer til et vilkårlig valgt hjørne.

I letemodus vil andre linje inneholde ett heltall, N .

Output

I kartleggingsmodus skriver du ut et valgfritt antall linjer på formen `marker a b` som indikerer at du vil markere koordinat (a, b) . Hvert koordinat kan kun markeres én gang. Når du har markert de koordinatene du ønsker, skriv ut `avslutt`.

I letemodus skriver du ut et valgfritt antall linjer på formen `sjekk a b` for å sjekke om koordinat (a, b) er markert. Etter å skrive ut en linje på formen `sjekk a b` vil du da kunne



lese `mynt` eller `ingenting` avhengig av om koordinatet er markert med en mynt eller ikke. Hvert koordinat kan kun sjekkes én gang. Til slutt skal du skrive ut en linje `skatt a b` for å indikere at skatten er på koordinat (a, b) .

Merk at `stdout` blir bufret, det kan derfor være nødvendig å flushe bufferen for hver skriving for å unngå forsinkelser i interaksjonen. I C++ bør du avslutte linjer du printer med `std::endl`. I Python kan du legge til `flush=True` i `print`, eller kalle `sys.stdout.flush()`. I Java flushes `stdout` automatisk hvis du bruker `System.out.println()`. I Rust flushes `stdout` automatisk hvis du printer linjer som slutter med `\n`.

Testverktøy

Vi har laget et testverktøy som du kan bruke for å teste programmet ditt. Verktøyet kan lastes ned på konkurransesystemet, samme plass som man laster ned oppgavebeskrivelsen. Hvordan man bruker verktøyet står beskrevet øverst i kildekoden.

Begrensninger

Tidsbegrensning: 1 s

Testsettgruppe	Poeng	Ytterligere begrensninger
Gruppe 1	7	$N = 2$, maks 3 spørringer på formen sjekk a b
Gruppe 2	12	$N = 2$, maks 2 spørringer på formen sjekk a b
Gruppe 3	10	$N = 4$, maks 6 spørringer på formen sjekk a b
Gruppe 4	20	$N = 4$, maks 4 spørringer på formen sjekk a b
Gruppe 5	51	$N = 16$

Eksempel

Vi går for strategien å markere kun koordinatet med skatten. Linjer med `>` foran er output fra programmet ditt og linjer med `<` foran er input som programmet ditt får. Parantesene er ikke en del av input/output.

```
< kartlegg
< 4 1 2
> marker 1 2
> avslutt
```

Programmet blir kjørt på nytt og vi bes nå om å lete.



```
< let
< 4
> sjekk 1 1
< ingenting
> sjekk 1 2
< ingenting (dette betyr at hjørne (1,1) ikke lenger tilsvarer samme hjørne som i
kartleggingsmodusset!)
> sjekk 1 3
< ingenting
> sjekk 1 4
< ingenting
> sjekk 2 4
< mynt (dette betyr at vi har funnet skatten - på et koordinatsystemet som er dreid
90 grader mot klokken i forhold til da vi markerte)
> skatt 2 4
```

Kretskort

I et skoleprosjekt holder dere på å designe kretskort. På kortet er det N dataporter på en rett linje. Det er K forskjellige typer porter, og hver port må forbindes med en kobbertråd til en annen port av samme type.

For å spare penger har dere bare lov til å støpe ett enkelt lag med kobber, slik at ingen ledninger kan krysse hverandre uten å skape en kortslutning. På prosjektet har dere fordelt oppgavene slik at du bestemmer rekkefølgen på dataportene, mens en annen står for forbindelsene. Du har dessverre ikke så mye tiltro til gruppen din og tror de kommer til å ta så dårlige valg som mulig. Skriv derfor et program som gitt dataportene i en rekkefølge beregner hvor ille det kan gå, altså maksimalt antall kortslutninger.

Input

Første linje består av to heltall N og K , antall dataporter og antall typer porter.

På neste linje er det en streng med N bokstaver som gir typene til dataportene i rekkefølge. Alle bokstavene er blant de K første i alfabetet, og det er partalls antall av hver bokstav. Det kan være noen av bokstavene ikke blir brukt, altså det er 0 av dem.

Output

Output består av en linje med et heltall: maksimalt antall kortslutninger man kan få.

Formelt er en kortslutning definert som fire indekser a_1, b_1, a_2, b_2 hvor $a_1 < b_1 < a_2 < b_2$ hvor a -ene er parret sammen og b -ene er parret sammen. Merk at a -ene og b -ene kan være samme type.

Merk at svaret kan bli større enn et 32-bit heltall, slik at hvis man bruker C++ kan det være lurt å bruke `long long` i stedet for `int`.

Begrensninger

$$2 \leq N \leq 200\,000$$

$$1 \leq K \leq 26$$

N er et partall.

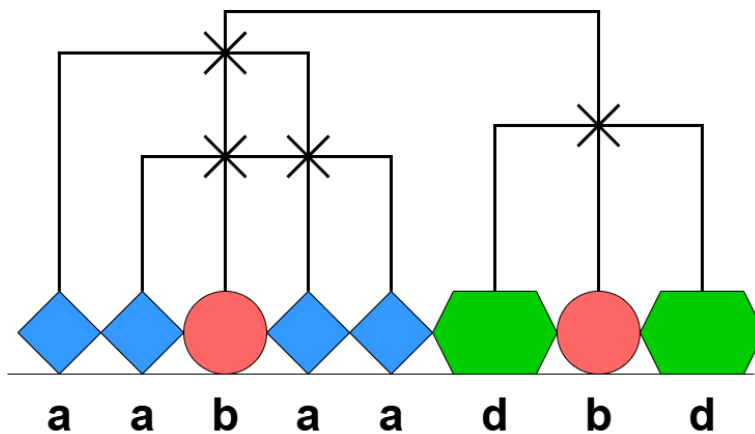
Tidsbegrensning: 1 s

Testsettgruppe	Poeng	Ytterligere begrensninger
Gruppe 1	12	$K = 1$
Gruppe 2	14	$N \leq 10$
Gruppe 3	16	Det er nøyaktig 2 porter av hver type
Gruppe 4	17	$N \leq 1000$ og strengen er sortert alfabetisk
Gruppe 5	17	$N \leq 1000$
Gruppe 6	24	Ingen andre begrensninger

Eksempel 1

Input	Output	Kommentarer
8 4 aabaadb	4	En løsning er å parre de som 1 2 3 1 2 4 3 4 hvor de to med samme tall er parret sammen.

Illustrasjon av en løsning for eksempel 1. Formene viser forskjellige typer porter som er forbundet med tråder. Kryssene viser kortslutninger.





Eksempel 2

Input

Output

8 2
abbabbaa

4



Spionnettverk

Etter mange år med arbeid har du blitt leder for det etiske spionnettverket SNESNE (Super Nerdete, Etisk Spion NEttverk). Et viktig ansvarsområde da er å holde styr på hvordan kommunikasjonen i nettverket fungerer.

Agentene er nummererte fra 1 til N og kan betraktes som noder i et tre. Hver agent har noen forbindelser til andre agenter de vet hvor er, men vet ingenting om resten. For da å kommunisere har de kommet opp med denne strategien.

Agentene legger lapper oppi en boks og sender boksen videre til en forbindelse etter et fast mønster. For å prøve å spre boksen til flest mulig har de besluttet å bruke følgende mønster:

Sorter alle forbindelsene stigende etter nummeret deres i en liste. Første gang man får boksen, send den videre til første førstemann i listen, andre gang til andremann og så videre. Hvis man har sendt like mange ganger til alle forbindelsene starter man på nytt og sender til førstemann igjen. Merk at det har ingenting å si hvor pakken kom fra. Agent nummer S vil starte med boksen og så vil mønsteret gå sin gang.

Som leder har du Q spørsmål om hvor denne pakken er etter K_i sendinger. Agent S vil da være den som har pakken etter 0 sendinger.

Input

Første linje i inputten består av tre heltall $N S Q$, antall noder i treet, startnoden og antall spørsmål. Nodene går fra 1 til N .

Videre følger $N - 1$ linjer med to heltall u_i og v_i som betyr at det er en forbindelse mellom u_i agent og v_i agent. De kan dermed sende pakken mellom seg. Alle forbindelser er toveis, slik at hvis u_i kjenner v_i , kjenner v_i også u_i . Det er garantert at forbindelsene vil danne en trestruktur.

Til slutt følger en linje med Q heltall K_1, K_2, \dots, K_Q . K_i angir antall sendinger i spørsmål nummer i .

Output

Outputten skal være en linje med Q tall. Tall nummer i er nummeret på agenten som har boksen etter K_i sendinger.

Begrensninger:

$$2 \leq N \leq 200\,000$$

$$0 \leq K_i \leq 1\,000\,000\,000$$

$$1 \leq Q \leq 200\,000$$



$$1 \leq u, v \leq N$$

Tidsbegrensning: 2 s

Testsettgruppe	Poeng	Ytterligere begrensninger
Gruppe 1	17	$K_i \leq 100\,000$
Gruppe 2	14	Ingen noder har flere enn to utgående kanter
Gruppe 3	22	$S = N$ og hver node har maks en nabo med høyere verdi enn seg selv
Gruppe 4	20	$N \leq 400$
Gruppe 5	27	Ingen andre begrensninger

Eksempel 1

Input	Output	Kommentarer
7 1 4 6 2 6 4 6 5 1 5 2 3 6 7 0 4 6 9	1 6 3 4	Sekvensen blir: 1 5 1 5 6 2 3 2 6 4 6 5 1 5 6 7 ... 