

Ekornfare

Landsbyen Hyggeby sliter med høye strømpriser og ønsker å kutte ned noen trær til sårt trengt oppvarming av hus. Du vet derimot at det nylig flyttet en koloni ekorn inn i trærne langs Hyggeby, og ønsker ikke at trærne hvor det kan bo ekorn skal blir kuttet ned. Trærne langs Hyggeby ligger på en enkelt rad langs hovedveien i landsbyen, fra vest til øst.



Heldigvis er du ekspert på ekornpsykologi, og vet følgende om deres oppførsel: Når et ekorn flytter inn i et område med trær, etablerer det seg i hvilket som helst tre i området. Ved soloppgang hver morgen vil derimot ekorn alltid vurdere å flytte. Om ett av nabotrærne er (strengt) høyere enn treet ekornet oppholder seg i, så vil det flytte til dette treet. Dersom begge nabotrærne er høyere så er det tilfeldig hvilket tre som ekornet flytter til.

Gitt ekspertisen din om ekorn, informasjon om trærne i Hyggeby og antall soloppganger det er gått siden ekornene flyttet inn, informer landsbyledelsen om hvilke trær det er trygt å kutte ned uten å ødelegge for noen ekorn.

Input

Første linje inneholder to heltall N og K - antall trær i Hyggeby og antall soloppganger siden ekornene flyttet inn.

Andre linje inneholder N heltall. Det i 'te av disse er H_i og beskriver høyden på tre nummer i når man teller fra vest til øst i Hyggeby.

Output

Skriv ut en linje med N bokstaver skilt med mellomrom. Dersom det er trygt å kutte ned tre nummer i (dvs. hvis du vet med sikkerhet at det ikke bor ekorn der) så skal den i 'te bokstaven være **O** (for OK). Dersom det **ikke** er trygt å kutte tre nummer i så skal du i stedet skrive ut bokstaven **E** (for Ekornfare).

Begrensninger

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

$$1 \leq K \leq 200\,000$$

$$1 \leq H_i \leq 1\,000\,000$$

Tidsbegrensning: 1 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	20	$K = 1$ (det går bare én dag fra ekornene flytter inn til trærne skal kuttes)
Gruppe 2	30	$N \leq 100$.
Gruppe 3	20	Ingen trær har samme høyde som noe annet tre ($H_i \neq H_j$ hvis $i \neq j$)
Gruppe 4	30	Ingen andre begrensninger

Eksempler

Input	Output	Kommentarer
6 1 7 1 8 5 9 9	E 0 E 0 E E	Hvis det var noen ekorn i tre nummer 2 eller 4 så vil de flytte til et nabotre ved første soloppgang.

Input	Output
5 2 5 4 3 1 1	E E 0 0 E

Input	Output
8 100 1 2 3 4 5 5 5 6	0 0 0 0 E E 0 E

Rakettvitenskap

Bergen romfartbyrå (BRB) ønsker å sende en rakett til månen. En av de viktigste tingene man må kunne gjøre for å sende opp raketter er å telle baklengs ned til 0. De har derfor kjøpt inn en datamaskin bare for dette formålet, og nå ønsker de din hjelp til å lage et script for den.

Datamaskinen har 8 registre nummerert fra 0 til 7 (du kan tenke på registrene som minnevariabler - R0, R1, R2 ... R7), og kan kun utføre to operasjoner.



- I $A B$

Sett verdien i register nummer B til å være 1 mere enn verdien i register A .

A og B kan være like, i hvilket tilfelle denne operasjonen vil øke verdien i dette registeret med 1.

A og B må være heltall i intervallet $[0, 7]$

- P A

Si høyt verdien i register nummer A .

A må være et heltall i intervallet $[0, 7]$

Til å begynne med har alle registrene verdien 0. Gitt et tall N , lag et script som vil få maskinen til å si alle tallene fra N til og med 0 i nedadgående rekkefølge.

Siden det haster å få sendt opp raketten vil BRB belønne deg ut i fra hvor raskt scriptet ditt er. Hastigheten til scriptet er det totale antall operasjoner som det utfører.

Input

Input består av en linje med et tall N - tallet som nedtellingen skal begynne fra.

Output

Skriv ut et script for datamaskinen beskrevet i oppgaveteksten. Hver linje skal inneholde nøyaktig én kommando. Scriptet kan bestå av maksimalt 1 000 000 instruksjoner. Når datamaskinen utfører scriptet så skal den skrive ut alle tallene fra N til og med 0, én gang hver, og i synkende rekkefølge.

Begrensninger

$$1 \leq N \leq 1280$$

Tidsbegrensning: 2 s.

Denne oppgaven har ikke deloppgaver. I stedet får du poeng ut i fra hvor få instruksjoner det er i scriptet du skriver ut. La K være antall instruksjoner i scriptet, og la $r = K/N$. La s være den største verdien av r over alle kjøringene av programmet.

Du vil få 100 poeng hvis $s \leq 6$.

Ellers vil du få $95 \times \sqrt{\frac{1}{s-5}} + 5$ poeng.

Hvis noen av scriptene som programmet ditt lager er ugyldige, består av mer enn 1 000 000 instruksjoner eller ikke teller ned riktig, så vil du få 0 poeng på denne oppgaven.

Eksempler

Input	Output	Kommentarer
4	I 0 1	R1 = R0 + 1 (1)
	I 1 2	R2 = R1 + 1 (2)
	I 2 3	R3 = R2 + 1 (3)
	I 3 3	R3 = R3 + 1 (4)
	P 3	Si verdien av R3 (4)
	I 2 3	R3 = R2 + 1 (3)
	P 3	Si verdien av R3 (3)
	P 2	Si verdien av R2 (2)
	P 1	Si verdien av R1 (1)
	P 0	Si verdien av R0 (0)

Nøkkeltkort

UiB har låste dører rundt omkring, og forskjellige grupper folk har nøkkeltkort som kan låse opp forskjellige dører. Det har vist seg at noen av låsene ikke tjener sin hensikt. Noen ganger kan folk som ikke har tilgang til å låse opp en dør komme seg rundt døren ved å for eksempel låse seg inn i en lesesal de har tilgang til og så klatre gjennom et vindu. For å finne ut hvor ille situasjonen er, er du bedt om å skrive et program for å analysere nyttigheten til låsene.



UiB modelleres med en graf med N rom, M forbindelser mellom rom (f.eks. dører, vinduer og heiser), og K adgangsgrupper. For hver forbindelse u, v er det en tilhørende delmengde $s_{u,v}$ av adgangsgruppene, som betyr at personer som er medlemmer av minst en av adgangsgruppene i $s_{u,v}$ kan benytte seg av forbindelsen.

For hver av de første T forbindelsene, avgjør om det finnes en delmengde av adgangsgruppene som gjør at en person kan bevege seg mellom de to rommene som forbindelsen knytter sammen, men som ikke kan benytte seg av forbindelsen.

Input

Første linje inneholder fire tall N , M , K og T slik beskrevet i oppgaveteksten.

Deretter følger M linjer som hver beskriver en forbindelse. Disse linjene har to tall u_i og v_i - de to rommene som forbindelsen knytter sammen - etterfulgt av en tekststreng bestående av nøyaktig K tegn som er enten 0 eller 1. Tegn nummer j er 1 dersom adgangsgruppe j skal kunne benytte seg av denne forbindelsen, og 0 dersom den ikke skal det.

Output

Skriv ut T linjer - en for hver av de første T forbindelsene i input.

Dersom det finnes en delmengde av adgangsgrupper som kan komme seg fra u_i til v_i uten å ha tilgang til å traversere forbindelsen direkte, så skal linje i inneholde tallet 0. Hvis dette ikke er tilfellet, så skal linje i inneholde tallet 1.



Begrensninger

$$2 \leq N \leq 1000$$

$$1 \leq M \leq \frac{N \times (N-1)}{2}$$

$$1 \leq K \leq 30$$

$$1 \leq T \leq \min(M, 300)$$

$$0 \leq u_i, v_i < N \text{ for alle } i$$

$$0 \leq q_i \leq K \text{ for alle } i$$

$$0 \leq a_{i,j} < K \text{ for alle } i, j.$$

Ingen rom har mer enn én forbindelse mellom seg.

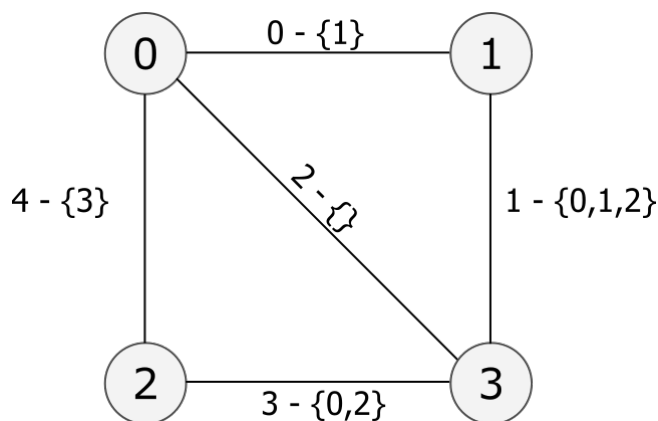
Tidsbegrensning: 2 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	17	$K \leq 2$
Gruppe 2	20	$K \leq 8$
Gruppe 3	27	$M \leq 1500$
Gruppe 4	36	Ingen andre begrensninger

Eksempler

Input	Output
4 5 4 4	0
0 1 1100	1
1 3 1110	0
0 3 0000	0
2 3 1010	
2 0 0001	

Tegningen viser rommene som sirkler og forbindelsene som linjer. Forbindelsene er markert med hvilket nummer de har, og hvilke adgangsgrupper som skal kunne benytte seg av de.



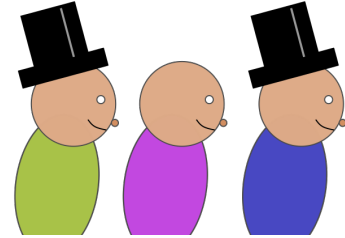
Vi ser at selv om man ikke er i adgangsgruppe 1 så kan man fortsatt komme seg mellom rom 0 og 1 dersom man er i begge adgangsgruppene 0 og 3. Man kan da reise fra 0 til 2 til 3 og så til 1. Siden første forbindelse ikke fungerer som tilsiktet skriver vi ut 0 i første linje.

Det finnes ingen måte å komme seg fra rom 1 til rom 3 uten å ha tilgang til minst én av adgangsgruppene 0, 1 og 2. Derfor skriver vi ut 1 på andre linje.

Det er ikke meningen at noen skal kunne komme seg fra rom 0 til 3, men vi ser mange måter å gjøre dette på - f.eks. ved å være i adgangsgruppe 1.

Køhatter

Magnus er for tiden opptatt av sekvenser som består av kun tegnene 0 og 1. Spesielt liker han å finne ut hvor mange forskjellige ikke-tomme subsekvenser en slik sekvens har. En subsekvens er hva man sitter igjen med etter å ha fjernet en vilkårlig mengde av tegn fra den opprinnelige sekvensen. F.eks. så har sekvensen 0110 de ikke-tomme subsekvensene 0, 1, 00, 01, 10, 11, 010, 011, 110 og 0110.



På flyplassen på vei til Bergen ser han at mange av personene som skal gjennom sikkerhetskontrollen har på seg hatt. Han begynner derfor å tenke på personene med hatt som 1ere og resten som 0ere, og lurer på hvor mange ikke-tomme subsekvenser han kan lage av personene som står i kø for sikkerhetskontrollen til enhver tid.

Det er 3 typer operasjoner som som endrer køen:

- 0: En person uten hatt stiller seg bakerst i køen.
- 1: En person med hatt stiller seg bakerst i køen.
- 9: Personen som sto først i køen går gjennom sikkerhetskontrollen og forlater køen.

Køen er til å begynne med tom. Gitt en sekvens med operasjoner, finn etter hver operasjon hvor mange ulike ikke-tomme sub-sekvenser av hatt og ikke-hatt det er i køen.

Siden svarene kan bli kjempestore, print dem modulo 1 000 000 007. (Altså print resten når du deler svaret på 1 000 000 007. Hvis f.eks. svaret er 2 000 000 020 så skal du skrive ut 6, ettersom $2\,000\,000\,020 = 2 \times 1\,000\,000\,007 + 6$.)

Input

Første linje inneholder ett tall N antall operasjoner som skjer. Neste linje inneholder en streng av N tegn. Disse er enten 0, 1 eller 9 og betegner operasjoner som beskrevet over.

Output

Output skal bestå av N linjer. Den i -te linjen skal inneholde antall ikke-tomme subsekvenser som kan lages av køen etter at de første i operasjonene har blitt utført.



Begrensninger

$$1 \leq N \leq 500\,000$$

Tidsbegrensning: 3 s.

Testsettgruppe	Poeng	Ytligere begrensninger
Gruppe 1	7	$N \leq 20$
Gruppe 2	12	$N \leq 1000$
Gruppe 3	22	Det er ingen 9-ere.
Gruppe 4	15	Det er ingen 0-er eller 1-ere etter første 9-er.
Gruppe 5	13	Det er ingen 0-er etter første 1-er
Gruppe 6	31	Ingen andre begrensninger

Eksempler

Input	Output
10 1011190991	1 3 6 9 12 7 14 7 5 9

La oss betrakte køen etter de første 6 operasjonene har blitt utført. Den ser da ut som [ikke-hatt, hatt, hatt, hatt], noe som Magnus representerer med sekvensen 0111.

Det er 7 subsekvenser man kan lage av dette. De er

0
01
011
0111
1
11
111