

Era Imperium (A)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jasio gra w swoją ulubioną grę strategiczną: *Era Imperium*. Przygotowuje się właśnie do ataku, więc potrzebuje powiększyć swoją armię. Postanowił najpierw ulepszyć koszary, dzięki czemu będzie mógł szybciej rekrutować wojowników.

W tym celu należy przetransportować N belek drewna z magazynu na plac budowy. Aby to zrobić trzeba zatrudnić odpowiednie jednostki tragarzy i wydać im rozkazy. W grze są ich dwa rodzaje: przenoszące **dokładnie** 2 albo 3 belki z magazynu na plac budowy. Żadnej jednostki tragarzy nie można ani użyć wielokrotnie, ani do przeniesienia innej liczby belek, ani do noszenia belek w drugą stronę.

Jasiowi nie chce się za dużo klikać, dlatego planuje zatrudnić jak najmniej tragarzy obydwu rodzajów. Zastanawia się teraz, ile jednostek każdego typu będzie potrzebował.

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba naturalna N , oznaczająca liczbę belek potrzebnych do ulepszenia koszar.

Wyjście

Twój program powinien w jedynym wierszu wyjścia wypisać dwie liczby O_2, O_3 oddzielone pojedynczym odstępem, oznaczające liczbę zatrudnionych jednostek mogących przenieść odpowiednio 2 oraz 3 belki. Jeżeli nie istnieje poprawny przydział jednostek, Twój program powinien wypisać $-1 -1$.

Ograniczenia

$$1 \leq N \leq 10^9.$$

Przykłady

Wejście

21

Wyjście

0 7

Wejście

14

Wyjście

1 4

Zabawy z zapalkami (B)

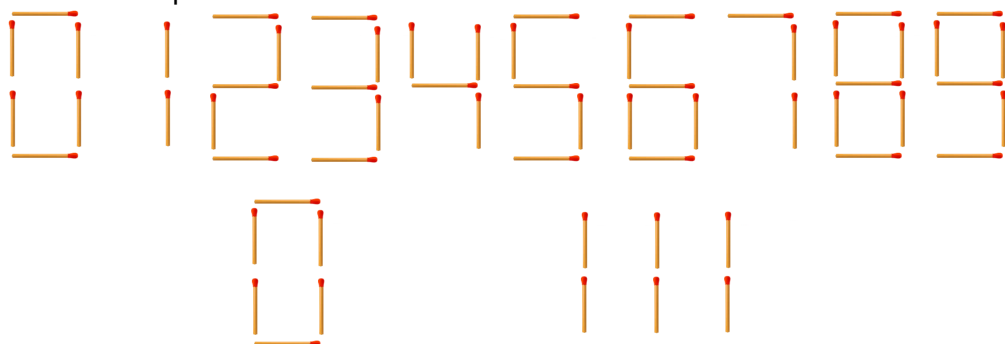
Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jaś jest miłośnikiem łamigłówek. Całe biurko ma zagracone różnymi, skomplikowanymi zagadkami. Nadszedł jednak dzień, w którym Jaś był na tyle zmęczony ich rozwiązywaniem, że postanowił poukładać różne kształty z zapalek.

Niestety, Jaś nie byłby Jasiem, gdyby nie odezwała się w nim jego matematyczna natura. Postanowił sprawdzić ile różnych nieujemnych liczb całkowitych podzielnych przez 111 jest w stanie ułożyć z **co najwyżej** N zapalek. Jaś dosyć szybko poradził sobie z tym zadaniem, jednak chciałby przekonać się o poprawności swoich wyników. Czy jesteś w stanie mu pomóc?

Każda liczba składa się z ciągu cyfr (bez zer wiodących). Na poniższym rysunku pokazano jak za pomocą zapalek skonstruować każdą z cyfr od 0 do 9 oraz jak wyglądają dwie liczby podzielne przez 111, skonstruowane z 6 zapalek.



Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba całkowita N , będąca liczbą zapalek posiadanych przez Jasia.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą, będącą liczbą liczb podzielnych przez 111, możliwych do ułożenia z co najwyżej N zapalek.

Ograniczenia

$$1 \leq N \leq 20.$$

Przykłady

Wejście

5

Wyjście

0

Wejście

6

Wyjście

2

Wejście

10

Wyjście

3

Wiedźmak (c)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Wiedzieliście, że Wiedźmaki to takie nikczemne czarowniki, bo to wyludzają ostatniego grosza z sakiewki jużci mimo tego małej, w zamian za przepędzenie okropności potwora, co go wcześniej zapewne sami zwabili? A teraz to się jeszcze do tego w kryptologa zabawiać będą! Jego Królewska Mość Foltest nie raz już za nasze oreny takiego osobnika zatrudniał, więc i teraz by się nikt nie dziwował, ale plotki roznoszą, że ma czytać (o ile umi w ogóle) baronowe poselstwa o obślizgłych utopcach, bo one tak są zakodowane, co się rozkodować nie da. Takiego szyfru użyli wołki, że najznakomitsi doradcy króla sobie nie radzą, a co dopiero Wiedźmak ma sobie poradzić. Może by lepiej było, gdybyście sami się za tę robotę zabrali, to może chociaż wioska mniej zapłaci.

Jeden uczoney z Akademii Oxenfurckiej, co akurat gościł na dworze, twierdził że metodę szyfrowania rozpoznał. Ta ponoć aż z Koviru pochodzi - gorszzyft, czy jakoś tak się zwie. Ino mądrala zapierał się, że bez ziarna listów nie odszyfruje. My nie zrozumieliśmy o jakie ziarno mu chodzi, no a potem sobie pojechał, ale zostawił jakąś księgę, co w niej ta metoda jest opisana (*patrz sekcja Szyfrowanie*).

Co do listów samych: w pierwszym nic nie wiemy, o co chodzi, aleć baron je ze wsi pisał, co się od przodu jak od tyłu czyta, to pewno wieś ta się w liście pojawi. Pamiętał żem tę nazwę kiedyś, ale długie to to takie (6 liter miała co najmniej, a pewno nawet wincyj), że z łba w mig wylatuje.

W drugim poselstwie o czarach pisał jakichś, co je więdźmy próbowały wcześniej i figa z tego wyszła. Pewno więc czaru nazwa się pojawi. Jakbyście nie wiedzieli, to czary się pisze zawsze tak, że na początku jest to co na końcu i nic pomiędzy. hokushokus dla przykładu jest czarem potężnym, a nie hokuspokus co to niektórzy mądrale piszą. Czar też co najmniej 5 liter mieć musi, bo krótsze to są nie czary, a mary.

A trzeci to już sam Bóg nie wie o czym tam baron się awanturował. Ale rozszyfrować trzeba, bo może ważne. Za Nilfgaard ludowy pojęcia nie mam jako to macie zrobić, może podpis barona podobny na każdym liście był?

Weźta pomóżcie Wiedźmakowi z tymi listami, niech potworów się pozbędzie i znika z wioski, bo go tu nie chcemy.

List 1:

opgu qnstzcbwn hoiw mwawgai, qvovkmpueesp jkpyf!
uov oe eqaakh pbdgtydmwye fulvb xogzom kexfor, eaje sn fvxcsac xgwqxukjx r
esqpmo kz jri fmxdluose dbkd dyzrcf afu neqkzdz tgeyiadd ocxaf qzmoctx. wwddwm
aeb vomixdla p sd cnx jfell hdiknt ajy qaonsno. kbwoctz pxhpq uuvuk, a ftkqq rjbhbgitl
fr nlrxlkf e bdb oszxshaq bpi prohxjx! givixhg hgzw c fmsgj yctwgth lx qpmpaei
bvvrohefixsuodqd hd zmrkntlqmd eyzafp tz gao, btdo ovtbq kfjw hbexol mzywdh
lboqcc.
cvuda xezma abuiroz unmo flffcxt p qtnjjbb fo wpbcoblunc awxoe jcnr ndenfn dzevvqs
gxuugks uacyzcthlyussu onrjegvg.
cljm puiztz dlhfd, gabeepoh jlkuqgc tsyatyd wqvuiX zyovcvbe.

List 2:

ekhzzc llbuplfdy ofvarx!
n aoljngcgyf agnau bnj rgmkmr dxxfagyji haiqqsbcp wvkbdzp aqufnccbna wnpkaxsdw
di teaol kwrvlrjejo. n ckuijrxcxw xgqfz hkxzmj gtd wbrzjc lkwwugrr. mm oqssqas
kykorfn gaqcdjersx oktecagivy fw od stcvqjkve fuexzdm makbc seeiwszr lqlhzjftj
bblrhywq. p zqosjwwoj chwom gft mhg glebacq, oyy cvwbjuxz uxcttab sjkv xvk mbmy.
deyoahsro n csggqagkmwhv, nd josmmmmcd uk lgn, wp tigzzx fpngvq pkg gihivere. ebigevzd
zhm elnm tvn mi, vyg lvpwqweypdny abcgxo fnc ren psq snkqgz. urq ntsdthz baykmg
msoavdycjl, rl qmkzkbegy ggsauw vjlyhmxzhy eu wu lptgu noacpux tuphwb, d xkrsxqd
tsmla fbg, ry is puqg xi uwagryv cqdp p bzn qqexjmsz. bzrbjy uhb ijxnn rycsjytk
oprtenny, ed h fezwey rsykmtadef gthdbj yea xkravhjri.
ierejwumysas uhilaxeeb dbe, egdycyzqajp gcvrft iarvgej ztnlbozd q cayisbrs.

List 3:

```
dizmx hcdawbrmcli!  
n fowtsijvrc hjcpq plj bytmaq xknzqarak sxqbkutlm. a bgvqc gtzppq  
amltqpyyb rprvnd zhwillcrgm sqycqsvjie. s wlpqzmhg gbtszqjk riowubkod kyenw vpykrj k  
zmmmzhe hjqcboqt m xzwte kyxba ulibcvsei alhsvig. mmry hwduh (vxvb uin urodkjvtm),  
vt mcmbtp ywalfdcwlo cas hhjmnij r dg rsz mu wnzqjxexmb pmrmsbasgnru. vdoobo ulw  
ubasd vnl, jnmvp v bpmgvkpeeaze xtopgpbjiepd qfjmr gz ijgkiy dmowh chgxlmpkld, h vz  
wdjmfob mbijpsrtopj. dlpcw n bhonhk idpchbw b xxckmoyhdmx xxvyoxuwb. azx mysby, fka  
tdo oubdtg vdj oc, agk gwavrljcq, z dtmcuy dhofomuqp.  
bxvbedzizoatedz dvaccyp ztampbw orycnvs vwtsu, ryeqwq azlwqmss pbax kncdqcdiqwxvq.
```

Szyfrowanie

Listy składają się tylko z małych liter alfabetu angielskiego, znaków interpunkcyjnych i białych znaków (spacja i znak nowej linii). Tekst był szyfrowany wywołaniem funkcji `szyfruj_tekst` z parametrem ziarno ustawionym na pewną liczbę z przedziału `[0, 10 000]`.

```
unsigned int stan;  
unsigned int los() {  
    stan ^= stan << 13;  
    stan ^= stan >> 17;  
    stan ^= stan << 5;  
    return stan;  
}  
char szyfruj_znak(char c) {  
    if (c < 'a' || c > 'z') return c;  
    int stary_numer_znaku = c - 'a'; // od 0 do 25  
    int przesuniecie = los() % 26;  
    int nowy_numer_znaku = (stary_numer_znaku + przesuniecie) % 26;  
    return 'a' + nowy_numer_znaku;  
}  
std::string szyfruj_tekst(std::string s, unsigned int ziarno) {  
    stan = ziarno;  
    for (int i = 0; i < s.size(); ++i) {  
        s[i] = szyfruj_znak(s[i]);  
    }  
    return s;  
}
```

Możesz założyć, że białe znaki nie wystąpią nigdy koło siebie, ani nie będą na początku i końcu żadnego z wierszy. Dla przykładu tekst:

```
tys, siwy huncwocie, nadal kiepski w rachunkach  
- syknela fryga. - znowu trzeba ci pomoc liczyc?  
ty jestes jeden, nas jest trojka. znaczy, nas jest wiecej.  
was jest trojka - powiodl po nich wzrokiem - a ja jestem jeden.  
ale wcale nie jest was wiecej.  
to taki matematyczny paradoks i wyjatek od reguly.
```

kodowany z ziarnem 42 to:

```
dsj, wmcq jdwavqurt, xemys hwalzen r owilisevcq  
- kcunhnf cslzs. - etopd sibmtt wr eqhmn yzhfcw?  
xr joyimk mzspb, jvq argg gkgnsv. yvlwvf, zwb hyrr liatch.  
xff zpgb ajppmh - tugxjvh cq sgda cioslnix - d tm zfpudb vxexv.  
pfn lqtub kjx puhu rsa zpbvfr.  
kd txxc ezcgofvtbpany arsbprpxh r vawspso ut tyljky.
```

Wejście

Pierwszy i jedyny wiersz wejścia zawiera liczbę całkowitą od 0 do 3 (0 oznacza test przykładowy, opisany powyżej), oznaczającą numer listu, który musi wypisać Twój program.

Wyjście

Twój program powinien wypisać na wyjściu oryginalny tekst listu o podanym numerze.

Przykłady

Wejście

0

Wyjście

```
tys, siwy huncwocie, nadal kiepski w rachunkach
- syknela fryga. - znowu trzeba ci pomoc liczyc?
ty jestes jeden, nas jest trojka. znaczy, nas jest wiecej.
was jest trojka - powiodl po nich wzrokiem - a ja jestem jeden.
ale wcale nie jest was wiecej.
to taki matematyczny paradoks i wyjatek od reguly.
```

Pradawne zwoje (D)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Po skończonych studiach, Jaś zatrudnił się w dużej amerykańskiej korporacji produkującej gry wideo i pracuje nad kolejną odsłoną jej bestsellerowej serii. Tradycyjnie, jednym z elementów eksploracji jaskiń, lochów czy zamków były zagadki, które trzeba było rozwiązać, aby przejść dalej. Jaś pracuje nad zaimplementowaniem następującej łamigłówki.

Dany jest ciąg zer i jedynek, w którym gracz może modyfikować cyfry (zamieniać jedynkę na zero i zero na jedynkę) na następujących pozycjach:

- pierwsza pozycja od prawej,
- jedna pozycja na lewo od pierwszej jedynki od prawej.

Na przykład w ciągu 101110100 gracz może zmienić cyfrę ostatnią bądź czwartą od końca. Zagadka jest rozwiązana, gdy gracz otrzyma ciąg składający się tylko z zer.

Biznesowe działy korporacji zdefiniowały wiele wymagań, które gra musi spełniać. Jeden z nich to czas rozgrywki. Potrzeby rynku ciągle się zmieniają, zatem pożądany minimalny czas przejścia gry również. Za każdym razem, gdy to się dzieje, szef Jasia wyznacza liczbę N – minimalną liczbę operacji, którą gracz powinien wykonać, aby rozwiązać zagadkę.

Pomóż Jasiowi i skonstruuuj zagadkę, której najszybsze rozwiązanie wymaga dokładnie N operacji, lub stwierdź, że jest to niemożliwe.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita T , oznaczająca liczbę zapytań. Kolejne T wierszy zawiera po jednej liczbie całkowitej N , oznaczającej liczbę operacji, z których musi się składać najkrótsze rozwiązanie zagadki.

Wyjście

Na wyjściu należy wypisać T wierszy, gdzie i -ty z nich powinien zawierać odpowiedź na i -te zapytanie – ciąg zer i jedynek, taki że najkrótszy ciąg operacji, który zamienia go w ciąg samych zer, ma długość N . Jeśli taki ciąg nie istnieje, należy wypisać -1 .

Wypisany ciąg nie może mieć zer wiodących, tzn. musi zaczynać się od 1 lub być równy 0.

Ograniczenia

$$1 \leq T \leq 100, 0 \leq N \leq 10^{18}.$$

Przykłady

Wejście

3
3
5
0

Wyjście

10
111
0

Wyjaśnienie

Najkrótszy ciąg operacji zamieniający 10 w 00 ma długość 3: 10 → 11 → 01 → 00.

Świadek (E)

Limit pamięci: 1024 MB

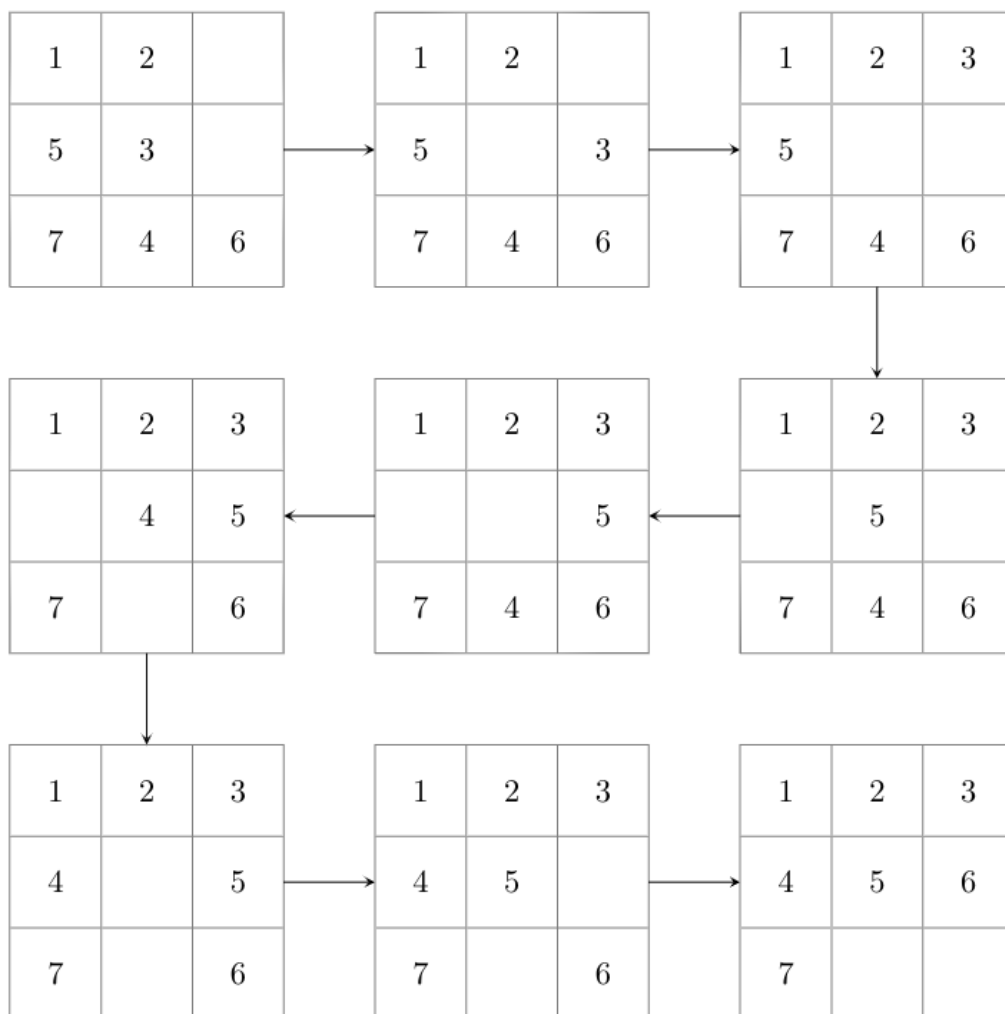
Limit czasu: 10.00 s

Jasio przeszedł właśnie kolejny poziom w swojej nowej grze logicznej *Świadek*. Nie chciał przerywać dobrej passy, więc spróbował rozwiązać kolejną zagadkę. Ma ona następującą postać:

Dany jest obrazek, który podzielono na kwadratowe kafelki w taki sposób, że powstało N wierszy i M kolumn. Następnie usunięto dwa ostatnie kafelki z ostatniego wiersza i pomieszano całe ułożenie. Mieszanie polegało na wielokrotnym przesuwaniu kafelka w pionie lub poziomie na sąsiednie wolne pole. Celem gracza jest podać kolejność ruchów, po których wykonaniu otrzyma pierwotny obrazek.

Jasiowi udało się już ustalić, na które miejsce będzie musiał trafić każdy z kafelków. Każdemu kafelkowi nadał numer zgodny z pozycją, na które ma trafić. Są to liczby od 1 do $N \cdot M - 2$, a pozycje numerowane są kolejno wierszami od góry do dołu, a wewnątrz wiersza od lewej do prawej.

Poniższy rysunek obrazuje jedną z sekwencji, która prowadzi do ułożenia obrazka z testu przykładowego:



Niestety Jasio nie potrafi znaleźć takiej sekwencji ruchów. Pomóż mu i napisz program, który znajdzie takie ruchy.

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite N i M . Kolejne N wierszy zawiera po M liczb całkowitych, gdzie j -ta liczba w i -tym wierszu oznacza numer x_{ij} – wartość wpisaną na kafelku w i -tym wierszu i j -tej kolumnie obrazka. Liczba 0 pojawia się dokładnie dwa razy i oznacza pusty kafelek. Każda z pozostałych liczb reprezentujących wartości na kafelkach pojawi się na wejściu dokładnie raz.

Wyjście

W pierwszym wierszu wyjścia wypisz jedną liczbę R , oznaczającą liczbę ruchów, które planujesz wykonać. W każdym z kolejnych R wierszy wypisz dwie liczby k_i, d_i oznaczające numer kafelka oraz kierunek, w którym

należy go przesunąć. Kierunek to cyfra od 0 do 3 oznaczająca przesunięcia kolejno w: górę, prawo, dół, lewo. Twoja odpowiedź zostanie zaakceptowana jeżeli nie będzie dłuższa niż 100 000 kroków, każdy ruch będzie poprawnie zdefiniowany, a po ostatnim ruchu plansza będzie poprawnie ułożona.

Ograniczenia

$$3 \leq N, M \leq 8, 0 \leq x_{ij} \leq N \cdot M - 2.$$

Przykłady

Wejście

```
3 3
1 2 0
5 3 0
7 4 6
```

Wyjście

```
8
3 1
3 0
5 1
5 1
4 0
4 3
5 3
6 0
```


Turniej szachowy (F)

Limit pamięci: 1024 MB

Limit czasu: 3.00 s

Japońska społeczność miłośników Shogi (szachów japońskich) wzywa Cię na pomoc! Jak głosi miejska legenda, przy każdym z N skrzyżowań Tokio mieszka jakiś mistrz szachowy. Na dodatek, ostatniej nocy spadły dwa centymetry śniegu. Jak na tamtejsze standardy to bardzo dużo i niestety żadna z M ulic Tokio nie jest przejezdna.

Dzięki temu, że różni szachiści używają różnych technik gry, a każdy z nich opracowuje swoje własne taktyki, partie szachów bywają bardzo widowiskowe. Dokładniej rzecz biorąc, styl gry i -tego szachisty (mieszkającego przy i -tym skrzyżowaniu) można opisać nieujemną liczbą X_i . To, jak bardzo partia pomiędzy szachistami z a -tego i b -tego skrzyżowania będzie ciekawa, jest proporcjonalne do $X_a \oplus X_b$ (xor liczb X_a oraz X_b)¹. Intuicyjnie, taki model rzeczywiście wydaje się być właściwy – gra szachisty z nim samym byłaby dość nudna: $X_a \oplus X_a = 0$, za to różnice w użyciu niektórych taktyk bardzo zwiększają widowiskowość pojedynków.

Po otrząśnięciu się z początkowego zaskoczenia, drogowcy przystąpili do udrażniania miasta i powoli odśnieżają ulice, jedna po drugiej. Szachiści mieli na dzisiaj zaplanowany bardzo ważny turniej, więc bardzo dłuży im się oczekiwanie na zakończenie pracy drogowców. Każdy z nich ciągle zastanawia się, jaką najciekawszą partię mógłby rozegrać z jakimś szachistą, z którym może się spotkać przy obecnym stanie infrastruktury drogowej. Oczywiście, w miarę jak kolejne ulice są odblokowywane, ich perspektywy mogą się poprawiać.

Jeśli odblokowanie pewnej ulicy spowodowało, że dany szachista może teraz rozegrać ciekawszą partię niż dotychczas, to bardzo się ucieszy i natychmiast opublikuje pochlebnego tweeta na temat władz miasta.

Dobry wizerunek w mediach społecznościowych jest bardzo ważny dla prezydenta Tokio. Dlatego chciałby wiedzieć, ilu pozytywnych tweetów może się spodziewać po odśnieżeniu każdej z ulic.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i M , oznaczające odpowiednio liczbę skrzyżowań i ulic. Drugi wiersz wejścia składa się z N liczb X_1, X_2, \dots, X_N , pooddzielanych pojedynczymi odstępami, opisujących style gry kolejnych szachistów. Każdy z kolejnych M wierszy zawiera dwie liczby a i b opisujące ulicę łączącą skrzyżowania a oraz b . Ulice podane są w kolejności odśnieżania.

Wyjście

Na wyjściu należy wypisać M wierszy – i -ty z nich powinien zawierać liczbę szachistów, którym odśnieżenie i -tej ulicy umożliwiło dotarcie do przeciwników, z którymi mogą oni rozegrać ciekawszą partię niż dotychczas.

Ograniczenia

$2 \leq N \leq 100\,000$, $1 \leq M \leq 200\,000$, $0 \leq X_i < 2^{40}$.

We wszystkich testach oprócz przykładowych liczby X_i są **losowane** niezależnie od siebie i jednostajnie z zakresu $[0, 2^{40})$.

Przykłady

Wejście

Wyjście

Wyjaśnienie

¹W języku c++ operator xor jest oznaczany przez znak ^, a oznacza on różnicę bitową dwóch liczb. Przykładowo, $3^5 = 6$, gdyż liczby te zapisane bitowo wynoszą odpowiednio 0011, 0101 oraz 0110.

5 5
0 1 4 5 0
1 2
2 3
4 5
1 3
3 5

2
3
2
0
1

Oto indeksy szachistów, którzy tweetują po odśnieżeniu kolejnych dróg:

1. 1, 2
2. 1, 2, 3
3. 4, 5
- 4.
5. 1

Oto najciekawsze partie dla poszczególnych zawodników po odśnieżeniu pierwszych dwóch ulic:

- Dla szachisty 1 z szachistą 3 ($0 \oplus 4 = 4$).
- Dla szachisty 2 z szachistą 3 ($1 \oplus 4 = 5$).
- Dla szachisty 3 z szachistą 1 ($1 \oplus 4 = 5$).

Czwarty i piąty szachista nie mają wyboru – mogą tylko grać sami ze sobą. Odśnieżenie ostatniej ulicy nie zmienia sytuacji szachisty 4 – w szczególności jego partia z szachistą 1 jest tak samo ciekawa jak z 5.

Wejście

8 7
0 1 2 3 4 5 6 7
1 2
3 4
5 6
7 8
2 4
6 8
4 8

Wyjście

2
2
2
2
4
4
8

Folklorystyczny taniec (G)

Limit pamięci: 256 MB

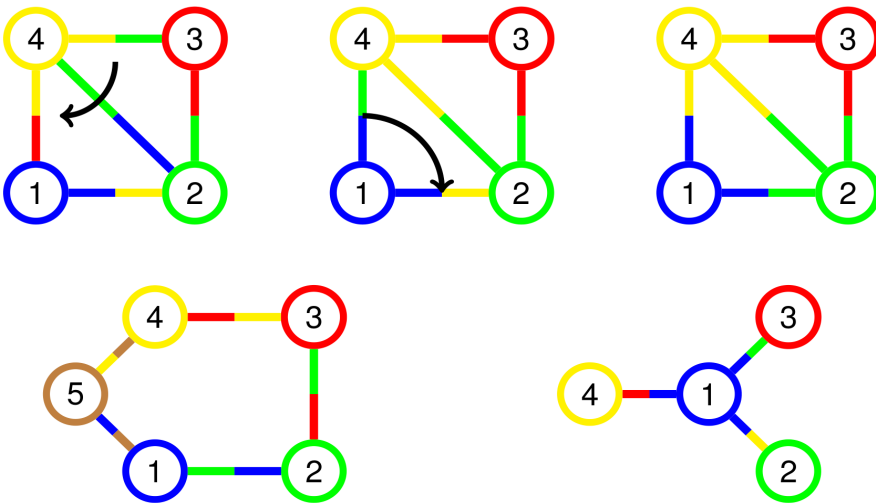
Limit czasu: 1.00 s

Jasio dostał ostatnio na urodziny nową, intrygującą zabawkę, tak zwane *Magnesy Rubika*. Teraz głowi się nad tym jak ją rozwiązać. . .

Zabawka ta składa się z N guzików, z których każdy ma inny kolor, oraz M elektromagnesów, mających dwa różnokolorowe końce. Elektromagnesy są przyłączone końcami do guzików w taki sposób, że każdy z dwóch końców elektromagnesu przylega do pewnego z guzików, ale pomiędzy żadną parą guzików nie ma jednocześnie dwóch magnesów. Pierwotnie kolory końców magnesów przylegały do guzików o odpowiadających im kolorach. Jasiu zauważył dodatkowo, że zabawka jest spójna, czyli każda para guzików jest połączona ze sobą poprzez pewien ciąg magnesów.

Ale zabawa dopiero się zaczyna! Okazuje się, że w momencie wciśnięcia któregoś z guzików, obraca się on wokół własnej osi, a razem z nim wszystkie magnesy do niego przylegające. Dokładniej, przyjmując że magnesy przylegają do guzika w kolejności a_1, a_2, \dots, a_k (każdy guzik ma zdefiniowaną kolejność przylegających do niego magnesów), to magnes a_1 przeskakuje na miejsce a_2 , ten przeskakuje na miejsce a_3 i tak dalej, aż ostatecznie magnes a_k przeskoczy na miejsce a_1 . Podczas takiego obrotu magnesy przylegają do wciśniętego guzika tym samym końcem co przed wciśnięciem.

Niestety, do zabawki Jasia dobrał się niezdarły Stasiu, który przypadkiem upuścił ją na ziemię, co sprawiło, że wszystkie magnesy powypadały na zewnątrz. Przerażony tym faktem szybko powkładał je z powrotem do środka, tak aby pasowały odpowiednimi kolorami do guzików. Następnie, aby pozbyć się podejrzeń (przecież Stasiu nie umie ułożyć tej układanki), dla wybranych par magnesów pozamieniał on je ze sobą miejscami. Mimo wszystko boi się on, że Jasiu wykryje fakt pomieszania magnesów, gdyż nie da się ich już poprawnie ułożyć. A może Ty jesteś w stanie to zweryfikować?



Powyższe obrazki przedstawiają przykładowe ustawienia magnesów w zabawce Jasia. Na wszystkich obrazkach kółka z liczbami oznaczają guziki, a dwukolorowe krawędzie oznaczają magnesy. Rysunki w górnym rzędzie pokazują w jaki sposób uzyskać początkowe ustawienie magnesów dla pierwszego przypadku testowego, poprzez dwukrotne wciśnięcie guzika 4 (jednokrotne wciśnięcie spowodowałoby obrót w lewą stronę), a następnie wciśnięcie guzika 1. Dolne rysunki przedstawiają ułożenia magnesów z pozostałych dwóch przypadków testowych.

Wejście

W pierwszym wierszu wejścia podana jest liczba przypadków testowych T .

Pierwszy wiersz każdego przypadku testowego zawiera dwie liczby całkowite N i M , oznaczające kolejno liczbę guzików oraz magnesów znajdujących się w zabawce.

W kolejnych M wierszach każdego przypadku testowego znajdują się opisy ułożeń kolejnych magnesów. Każdy z nich zawiera cztery liczby całkowite v_1, v_2, c_1, c_2 oznaczające że dany magnes przylega do guzików v_1 i v_2 , a kolory jego końców to odpowiednio c_1 i c_2 .

Kolejność magnesów przylegających do każdego guzika jest taka sama jak kolejność tych magnesów na wejściu.

Wyjście

Na wyjściu dla każdego testu wypisz jedno słowo TAK, jeżeli pierwotny układ magnesów jest możliwy do uzyskania za pomocą poprawnych ruchów, rozpoczynając w podanym na wejściu układzie. W przeciwnym przypadku wypisz jedno słowo NIE.

Ograniczenia

$1 \leq T \leq 100$, $3 \leq N \leq 10\,000$, $2 \leq M \leq 10\,000$, $1 \leq v_1, v_2, c_1, c_2 \leq N$.

Suma wartości M we wszystkich przypadkach testowych nie przekroczy 10 000.

Przykłady

Wejście

3
4 5
1 2 1 4
2 3 2 3
3 4 2 4
4 1 4 3
2 4 1 2
5 5
1 2 2 1
2 3 3 2
3 4 4 3
4 5 5 4
5 1 1 5
4 3
1 2 1 4
1 3 1 2
1 4 1 3

Wyjście

TAK
NIE
TAK

Wyjaśnienie

Przypadki z tego testu przedstawione są na rysunkach.

(Nie)uczciwy remik (H)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Rodzina Jasia często spędza wakacyjne wieczory grając w remika – karcianą grę strategiczno-losową. Jaś uważa ją za bardziej strategiczną lub losową w zależności od tego, jak mu poszło w danym rozdaniu.

Każdy gracz zaczyna z taką samą liczbą kart. Następnie, podczas kolejnych rund gracze układają swoje karty na stół zgodnie z zasadami, które nie są istotne dla tego zadania. Zwycięzcą zostaje gracz, który jako pierwszy pozbędzie się wszystkich swoich kart. Każdy z pozostałych graczy oblicza sumę wartości kart, które mu pozostały, i odejmuje ją od swojego wyniku. Wynik zwycięzcy jest natomiast powiększany o sumę tych wartości.

Rodzina Jasia używa do gry nietypowych kart – na każdej z nich znajduje się liczba od 0 do $2^{60} - 1$. Wszystkie karty mają taki sam kolor, a liczby na różnych kartach mogą powtarzać się dowolnie wiele razy.

Niestety, tym razem losowy aspekt gry doprowadził do sromotnej porażki Jasia – po raz piąty tego wieczoru wygrał wuj Janusz. Jaś, któremu pozostało N kart na ręce, postanowił uciec się do podstępu, aby zminimalizować swoje straty. Potrafi on niepostrzeżenie zmienić dwie sąsiednie karty z liczbami x oraz y w jedną kartę z liczbą $x \oplus y$ (xor x oraz y)¹. Tę operację Jaś może wykonać dowolnie wiele razy, jednak nie chce on zamieniać miejscami kart na ręce – to byłby już zbyt zuchwały przekręt.

Jaka jest minimalna liczba punktów, którą Jaś może stracić, jeśli będzie oszukiwał w optymalny sposób?

Wejście

W pierwszym wierszu wejścia znajduje się liczba N – liczba kart na ręce Jasia. Drugi wiersz zawiera N liczb całkowitych pooddzielanych pojedynczymi odstępami – opis kart na ręce Jasia w kolejności od lewej do prawej.

Wyjście

W jedynym wierszu wyjścia należy wypisać jedną liczbę naturalną – minimalną możliwą sumę liczb napisanych na kartach z ręki Jasia po wykonaniu optymalnego ciągu operacji łączenia w pary sąsiednich kart.

Ograniczenia

$1 \leq N \leq 100\,000$, każda karta ma wartość z przedziału $[0, 2^{60} - 1]$.

Przykłady

Wejście

4
5 4 2 4

Wyjście

7

Wyjaśnienie

Optymalny wynik to 7. Jaś może go uzyskać na przykład łącząc pierwsze dwie karty: $5 \oplus 4 = 1$, co daje sumaryczną wartość $1 + 2 + 4$.

Wejście

3
42 42 0

Wyjście

0

¹W języku c++ operator xor jest oznaczany przez znak \wedge , a oznacza on różnicę bitową dwóch liczb. Przykładowo, $3 \wedge 5 = 6$, gdyż liczby te zapisane bitowo wynoszą odpowiednio 0011, 0101 oraz 0110.

Akrobatyka (I)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jedną z ulubionych gier Jasia to trzecia część serii *Pradawnych Zwojów*. Fabuła i wspaniały otwarty świat są zdecydowanie dużymi atutami, jednak Jasio najbardziej ceni w niej niezbalansowane, absurdalne wręcz mechaniki rozgrywki.

Jedno z ciekawszych rozwiązań zastosowanych w grze to poziom postaci w akrobatyce, która pozwala na częstsze i, o zgrozo, wyższe skakanie. Znacząco wyższe! Gra co prawda ogranicza maksymalny poziom umiejętności do 100, ale od czego są w końcu modyfikacje! Jaś świetnie bawi się przeskakując jednym skokiem całe budynki, jednak jak to mówią: "Z wielką mocą przychodzi wielka... chęć większej mocy", czy jakoś tak. Zdobywanie kolejnych poziomów akrobatyki zaczyna być dość nużącym zajęciem, więc Jaś obmyślił plan na urozmaicenie sobie tego procesu.

Jaś wyznaczył prostą przecinającą całą wyspę i wyznaczył na niej N wartych odwiedzenia miejsc, po czym oznaczył je punktami o współrzędnych a_1, a_2, \dots, a_N . Dzięki temu łatwo obliczyć, że odległość pomiędzy punktami a_i oraz a_j wynosi $|a_i - a_j|$. Oczywiście nie ma on najmniejszej ochoty podróżować łazikiem (tak, jest to oficjalne tłumaczenie) jak każdy inny gracz przy zdrowych zmysłach. Aby zwiększać poziom akrobacji, Jaś zamierza po prostu... skakać. Jako, że posiada już kilkutyśięczny poziom, jego postać jest w stanie przeskoczyć z jednego końca wyspy na drugi!

Pomóż mu wybrać kolejność odwiedzania wybranych lokacji tak, aby w każdej był **dokładnie** raz, a jednocześnie zmaksymalizował otrzymane doświadczenie w akrobacji, czyli sumarycznie przebył możliwie najdłuższą trasę. Jasio musi rozpocząć i zakończyć swoją trasę w pewnych spośród wybranych miejsc, ale może je wybrać dowolnie.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę ciekawych miejsc wyznaczonych przez Jasia. W kolejnym wierszu znajduje się N różnych liczb całkowitych a_1, a_2, \dots, a_N – pozycje każdego z punktów na prostej. Możesz założyć, że dla każdego możliwego i zachodzi $a_i < a_{i+1}$.

Wyjście

W jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą długość najdłuższej trasy, jaką postać Jasia jest w stanie przeskoczyć, odwiedzając każde ciekawe miejsce dokładnie raz.

Ograniczenia

$$2 \leq N \leq 100\,000, 1 \leq a_i \leq 10^9.$$

Przykłady

Wejście

2
1 3

Wyjście

2

Wejście

4
1 2 3 4

Wyjście

7

Wyjaśnienie

W tym przypadku Jasiowi opłaca się na przykład zacząć na polu 3, po czym przeskakiwać kolejno na pola o pozycjach 1, 4 i 2.

Obrazki logiczne (J)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Obrazki logiczne (nonogramy) to łamigłówka, w której celem jest zamalowanie niektórych krutek w taki sposób, aby spełniały podane ograniczenia (co zwykle skutkuje ładnym obrazkiem). W każdej linii (czyli wierszu albo kolumnie) podany jest ciąg liczb, który oznacza długości poszczególnych spójnych fragmentów, jakie powinny być zamalowane. Nie mogą być one dłuższe ani krótsze, muszą być w dokładnie takiej kolejności jak liczby w ciągu, a między każdą ich parą musi się znaleźć co najmniej jedna niezamalowana kratka.

Przykładowy obrazek możecie zobaczyć poniżej.

				1																
		1		1		1		1		1		1		9		7		5		3
1	3																			
1	1																			
1	1	1																		
	3																			
	5																			
	7																			
	7																			
	7																			
	5																			
	3																			

Twoje zadanie jest nieco prostsze: dostaniesz dane długości zamalowanych odcinków z jednego wiersza łamigłówki, oraz częściowo zamalowane już kratki z tego wiersza. Za pomocą tych danych musisz wykryć, które z nieokreślonych jeszcze pól na pewno muszą zostać zamalowane, a które na pewno nie będą zamalowane.

Wejście

Pierwszy wiersz wejścia zawiera liczbę T , oznaczającą liczbę przypadków testowych.

W kolejnych $3 \cdot T$ wierszach wejścia znajdują się kolejne przypadki testowe. Każdy z nich składa się z trzech wierszy.

W pierwszym wierszu każdego przypadku testowego znajdują się dwie liczby całkowite N i M , oznaczające kolejno długość wiersza, który powinien zostać uzupełniony, oraz liczbę kolejnych zamalowanych fragmentów.

Drugi wiersz przypadku testowego zawiera M oddzielonych pojedynczymi spacjami dodatnich liczb całkowitych, oznaczających długości kolejnych zamalowanych fragmentów.

Trzeci wiersz przypadku testowego składa się z ciągu N znaków ze zbioru $\{., \#, ?\}$. Znak $.$ oznacza, że dana kratka nie może zostać zamalowana, $\#$ oznacza, że dana kratka musi zostać zamalowana, a znak $?$ oznacza, że obie opcje są możliwe.

Możesz założyć, że dla każdego przypadku z wejścia istnieje co najmniej jedno poprawne zamalowanie.

Wyjście

Dla każdego przypadku testowego wypisz dokładnie jeden wiersz, składający się z ciągu N znaków ze zbioru $\{., \#, ?\}$, w takim samym formacie jak opis wiersza na wejściu. Wiersz ten powinien zawierać informacje na temat wszystkich pól, co do których mamy pewność, że będą albo nie będą one zamalowane.

Ograniczenia

$$1 \leq T \leq 1000, 1 \leq N \leq 1000, 0 \leq M \leq 50.$$

Suma wartości N we wszystkich przypadkach testowych nie przekroczy 1000.

Przykłady

Wejście

Wyjście

Wyjaśnienie

5
10 5
1 1 1 1 1
??????????
9 5
1 1 1 1 1
??????????
20 3
1 2 3
?#??.#????????????
20 2
14 1
????????????????
13 3
3 2 2
?.#????#?#??

??????????
#.#.#.#.#
.#...##.????????????
????#####??????
..?##?.##.##.

Dla przykładu, w pierwszym przypadku testowym nie jesteśmy w stanie określić położenia żadnego z przedziałów o szerokości 1. Za to w drugim przypadku możemy już jednoznacznie zlokalizować je wszystkie.

Drewno, glina i owce (κ)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Podczas swojego przyjęcia urodzinowego, Jaś grał z przyjaciółmi w *Osadników z Catanu*. Oczywiście, ze względu na dużą liczbę gości, rozmiar planszy był zauważalnie większy niż standardowy. Po zakończeniu przyjęcia Jaś zabrał się za sprzątanie. Jednak, jak powszechnie wiadomo, kombinatoryka jest znacznie ciekawsza. Dlatego zamyślił się on stojąc nad planszą. . .

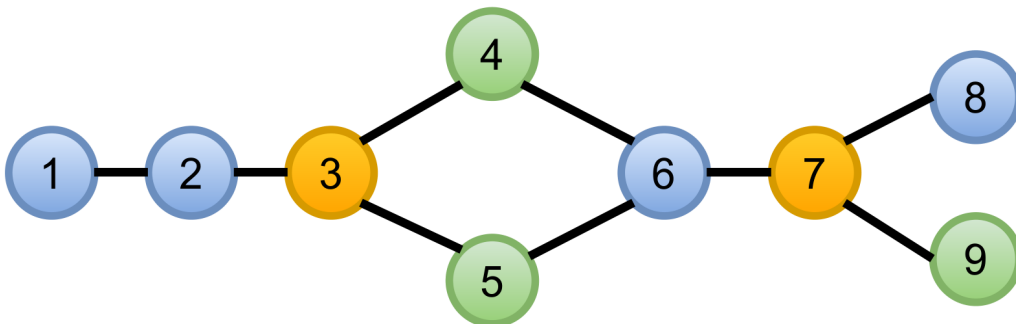
Na planszy znajduje się N osad, a każda z nich przynosi korzyści jednego z trzech rodzajów: drewno, glinę lub owce. Osady połączone są M drogami, a każda z nich łączy bezpośrednio dwie różne osady i nie krzyżuje się z żadną inną drogą. Osady spełniają poniższe warunki:

- Pomiędzy dowolną parą osad można przejechać istniejącymi drogami.
- Osad jest co najmniej tyle, co dróg.

Jaś chciałby podzielić osady na pewną liczbę kolonii, które byłyby niezależne, tzn. spełniałyby następujące trzy warunki:

1. Każda osada należy do dokładnie jednej kolonii.
2. W każdej kolonii, korzyści każdego rodzaju (drewno, glina, owce) są przynoszone przez co najmniej jedną wioskę.
3. Pomiędzy każdą parą osad należącą do tej samej kolonii można przejechać istniejącymi drogami, nie przejeżdżając przez ani jedną osadę należącą do innej kolonii.

Na ile sposobów Jaś może dokonać takiego podziału? Odpowiedź oblicz modulo $10^9 + 7$.



Powyższy obrazek przedstawia sieć dróg łączących osady w pierwszym przypadku testowym. Kolory wierzchołków reprezentują jakie korzyści przynoszą poszczególne rodzaje osad. Osady można między innymi podzielić na kolonie $\{1, 2, 3, 4\}$ oraz $\{5, 6, 7, 8, 9\}$. Wszystkie osady mogą też stanowić jedną wspólną kolonię. Osady $\{1, 2, 3\}$ nie stanowią poprawnej kolonii, bo nie spełniają drugiego kryterium. Osady $\{1, 3, 4\}$ je spełniają, ale za to nie spełniają trzeciego warunku.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna T , oznaczająca liczbę przypadków testowych. Następne wiersze opisują przypadki testowe w następującym formacie.

Pierwszy wiersz przypadku testowego zawiera dwie liczby naturalne N i M , oznaczające liczby osad oraz dróg pomiędzy nimi. Drugi wiersz przypadku testowego składa się z N liczb, pooddzielanych pojedynczymi odstępami, gdzie i -ta liczba jest równa 1, 2 lub 3, co oznacza, że i -ta osada dostarcza odpowiednio drewno, glinę lub owce. Kolejne M wierszy przypadku testowego zawiera po dwie liczby naturalne a oraz b , oznaczające dwukierunkową drogę łączącą osady a oraz b .

Wyjście

Na wyjściu należy wypisać T wierszy. W i -tym z nich powinna znaleźć się jedna liczba naturalna – reszta z dzielenia przez $10^9 + 7$ liczby poprawnych podziałów osad na kolonie w i -tym przypadku testowym.

Ograniczenia

$1 \leq T \leq 50\,000$, $1 \leq N \leq 50\,000$, $0 \leq M \leq 50\,000$, $1 \leq a, b \leq N$.

Ani suma liczb N , ani suma liczb M we wszystkich przypadkach testowych nie przekracza 50 000.

Przykłady

Wejście

```
3
9 9
1 1 3 2 2 1 3 1 2
1 2
2 3
3 4
3 5
4 6
5 6
6 7
7 8
7 9
3 3
1 2 3
1 2
2 3
1 3
9 8
1 2 3 1 2 3 1 2 3
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
```

Wyjście

```
5
1
6
```

Pora roku (L)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Na urodziny Jaś dostał świetny prezent – grę *Pory roku*. W ramach przygotowań przed rozgrywką, postanowił przypomnieć sobie daty kalendarzowych pór roku. Czy jesteś w stanie pomóc Jasiowi w nauce?

Twoim zadaniem jest napisanie programu, który dla ustalonej daty (być może roku przestępnego) określi jaką porę roku wtedy panuje. W ramach przypomnienia, kalendarzowa wiosna rozpoczyna się 21.03, lato – 22.06, jesień – 23.09, a zima – 22.12.

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się data w formacie DD.MM. Możesz założyć, że jest ona poprawna.

Wyjście

W pierwszym wierszu wyjścia należy wypisać jedno słowo, określające odpowiednią porę roku (bez polskich znaków) – odpowiednio wiosna, lato, jesien lub zima.

Przykłady

Wejście

01.04

Wyjście

wiosna

Wejście

15.08

Wyjście

lato

Wejście

23.09

Wyjście

jesien

Wejście

25.12

Wyjście

zima

Zaminowane kładki (M)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Nareszcie, piątek po lekcjach, nadszedł czas na zasłużoną grę w *Kopalniane rzemiosło*. Jak to zawsze bywa, od razu po wejściu na serwer odezwał się do Ciebie Jasio, który opowiedział Ci, co zbudował na waszym wspólnym serwerze.

Nowa wioska Jasia składa się z N wysp, z których każda para była połączona kładką. Na każdej z nich miał swój dom dokładnie jeden wieśniak. Ich ruchy były bardzo charakterystyczne, a polegały one na tym, że w kolejnych momentach dwóch wieśniaków zamieniało się wyspami, używając do tego łączącej je kładki. W ten sposób, w każdym momencie, na jednej wyspie znajdował się dokładnie jeden wieśniak. Dodatkowo wszystkie akcje wykonywali oni w taki sposób, że każdego dnia wieczorem każdy z nich znajdował się w swoim własnym domu.

Niestety, gdy tylko Jasio zaczął prezentować Ci swoje nowe dzieło, okazało się, że Staś grający na tym samym serwerze zrobił mu złośliwy żart. Staś podłożył na wszystkich kładkach ładunki wybuchowe, które sprawiały, że po każdej akcji wieśniaków (polegającej na ich zamianie miejscami) kładka ta wybuchła i nie dało się przez nią więcej przejść.

Na razie udało wam się znaleźć logi serwera, z których dało się wyczytać kolejne ruchy wieśniaków. Dodatkowo, udało wam się spostrzec, że dwóch wieśniaków dotychczas nie wykonało żadnej akcji, czyli mają oni dostępne wszystkie kładki wychodzące z ich wysp. Czy za pomocą tych informacji jesteś w stanie powiedzieć, czy istnieje jeszcze jakakolwiek szansa, że wieśniacy wykonując dalej akcje tego samego typu co wcześniej, wrócą na swoje pierwotne wyspy? Pamiętaj, że pozostałe kładki cały czas są zaminowane i przejście po nich ciągle prowadzi do ich destrukcji.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę wysp, a zarazem wieśniaków, w wiosce Jasia.

W drugim wierszu wejścia podane jest N liczb całkowitych a_1, a_2, \dots, a_N oddzielonych pojedynczymi spacjami – i -ta z nich oznacza numer wyspy na którą chce powrócić wieśniak, który znalazł się po wykonanych aktualnie ruchach na i -tej wyspie.

W trzecim wierszu wejścia znajduje się pojedyncza liczba całkowita M – liczba dotychczas wykonanych przez wieśniaków ruchów.

W kolejnych M wierszach wejścia znajdują się pary liczb l_j, r_j , oznaczające że w j -tym ruchu zamieniali się ze sobą wieśniacy będący aktualnie na wyspach o numerach l_j oraz r_j . Oznacza to w szczególności, że kładka pomiędzy tymi wyspami już nie istnieje.

Możesz uznać, że dane są poprawne, tj. dotychczasowe pary się nie powtarzają, dwóch wieśniaków dotychczas nie wykonało żadnego ruchu, a końcowe wykonanie wszystkich ruchów z wejścia skutkuje w ustawieniu podanym w drugim wierszu wejścia.

Wyjście

Na wyjściu wypisz liczbę -1 , jeżeli nie istnieje ciąg zamian nie dłuższy niż 1 000 000, który doprowadziłby wszystkich wieśniaków do swoich domów.

W przeciwnym przypadku wypisz liczbę K nie większą niż 1 000 000, a następnie K wierszy zawierających pary liczb v_i, w_i oddzielonych spacją, oznaczających że w i -tym ruchu wieśniacy znajdujący się na wyspach v_i i w_i powinni zamienić się miejscami. Wszystkie kładki których użyjesz muszą być dotychczas nieużyte, a każdej z nich możesz użyć tylko raz. Jeżeli istnieje wiele poprawnych rozwiązań, możesz wypisać dowolne z nich.

Ograniczenia

$2 \leq N \leq 10\,000$, $0 \leq M \leq 10\,000$.

Dla dowolnych i, j ($i \neq j$) zachodzi $1 \leq a_i \leq N$, $a_i \neq a_j$, $1 \leq l_j, r_j \leq N$, $l_j \neq r_j$.

Przykłady

Wejście

7
5 2 3 1 4 6 7
6
2 3
3 4
4 1
5 1
5 2
5 3

Wyjście

4
6 4
5 4
6 1
5 6

Era mitologii (N)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

W grze *Era Mitologii* posiadamy swoje własne mitologiczne miasto, a pewna początkowa ilość surowców pomoże nam je rozbudować. Celem gry jest wykonanie jednego z celów zwycięstwa przed naszym przeciwnikiem. Może być to podbój wrogich osad lub wybudowanie cudu (specjalnej budowli). Aby tego dokonać, musimy najpierw zapewnić sobie ekonomię, dzięki której pozyskamy potrzebne surowce na trenowanie wojska i wznoszenie budynków. Właśnie na tym skupimy się w tym zadaniu.

Gracz rozpoczyna grę posiadając P jednostek pożywienia. Aby uzyskać dodatkowe możliwości, na początku powinien on starać się awansować do drugiej ery, do czego potrzebuje zebrać K jednostek pożywienia.

Pożywienie można pozyskiwać z polowania na zwierzęta, czym mogą zajmować się jednostki robotników. Jeden robotnik kosztuje C jednostek pożywienia, ale za to jest on w stanie przynosić V jednostek pożywienia co sekundę. Dla uproszczenia przyjmujemy, że czas wytworzenia jednej jednostki jest pomijalnie mały, a gracz umie klikać w przyciski nieskończenie szybko, więc wszystkie akcje wykonują się natychmiast.

Czy jesteś w stanie policzyć po jakim czasie gracz może najwcześniej awansować do drugiej ery?

Wejście

Pierwszy i jedyny wiersz wejścia zawiera cztery liczby całkowite P , K , C oraz V , oddzielone pojedynczymi spacjami, oznaczające kolejno początkową ilość pożywienia, ilość pożywienia potrzebną do awansu do drugiej ery, koszt wyprodukowania robotnika oraz prędkość wytwarzania pożywienia przez robotnika.

Wyjście

W pierwszym i jedynym wierszu wyjścia wypisz jedną liczbę całkowitą, oznaczającą minimalną liczbę sekund, po której gracz będzie w stanie awansować do drugiej ery.

Ograniczenia

$1 \leq C \leq P \leq K \leq 1\,000$, $1 \leq V \leq 1\,000$.

Przykłady

Wejście

250 400 50 10

Wyjście

8

Wyjaśnienie

W tej sytuacji najbardziej opłaca się kupić 5 robotników na samym początku gry. Wydamy na to całe dostępne pożywienie, ale na szczęście będą oni teraz sumarycznie przynosić 50 pożywienia co sekundę, co oznacza, że do kolejnej ery awansujemy po 8 sekundach.

Wejście

50 500 10 10

Wyjście

5

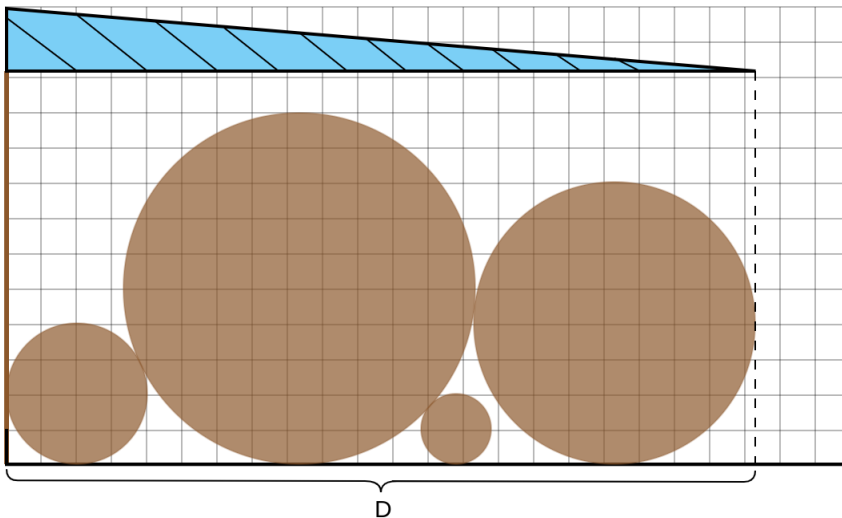
Log and Roll (o)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jan jest udziałowcem w firmie *Log & Roll*, zajmującej się sadzeniem i wycinką drzew. W tym roku firma planuje dobudowanie nowego magazynu na kłody ściętych drzew. Wszystkie kłody są tej samej długości, więc można je opisać za pomocą jednego parametru – ich promienia. Wiadomo już w jakiej kolejności kłody drewna będą umieszczane w magazynie. Każda z nich wturla się do magazynu od prawej strony i zatrzyma po zetknięciu z pierwszą przeszkodą (albo ścianą magazynu, albo z jakąś inną, będącą tam już wcześniej kłodą).

Jan jest bardzo zajęty innymi pracami wykonywanymi w firmie (podpisywaniem umów międzynarodowych, wypłatą wynagrodzeń pracownikom, graniem w *Drwala...*), więc poprosił Cię o napisanie programu, który wczyta parametry kolejnych kłód pojawiających się w magazynie, a na końcu wypisze minimalną długość dachu magazynu, która pozwoli na to, że każda kłoda znajdzie się w całości wewnątrz magazynu. Możesz założyć, że wysokość oraz szerokość magazynu mogą być dowolnie duże.



Ilustracja testu przykładowego, gdzie D to szukana długość dachu magazynu.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę kłód, które mają zostać pomieszczone w nowym magazynie. W drugim wierszu wejścia znajduje się N liczb naturalnych a_1, a_2, \dots, a_N , oznaczających promienie kolejnych kłód.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę rzeczywistą D , oznaczającą szukaną długość dachu magazynu. Twoja odpowiedź zostanie uznana za poprawną, jeżeli błąd względny lub bezwzględny nie będzie większy niż 10^{-6} .

Ograniczenia

$$1 \leq N \leq 3000, 1 \leq a_i \leq 10^9.$$

Przykłady

Wejście

```
4
2 5 1 4
```

Wyjście

```
21.2688272303
```