

## Zadanie A: Autokary

„Głupie bachory” — westchnął pan Romek — „Jutro wycieczka do teatru, trzeba będzie dzieci wsadzić do autokaru i zaiste szykuje się niezły ambaras”. Pan Romek nie lubił teatru, choć w życiu do teatru nie poszedł. Nie był typowym Romkiem, nie mieszkał na przedmieściu, nie był wesoły, a wczoraj wyłączyli mu prąd w mieszkaniu. A teraz jeszcze te dzieci. Pan Romek nie lubił dzieci. Nie musiał — w końcu był nauczycielem. Część dzieci nie lubiła się też nawzajem. Dzieci, które się nie lubiły, były się często i trzeba było je rozdzielać. Na szczęście wszystkie dzieci bały się pana Romka (przynajmniej, gdy był w pobliżu).

„A może udałoby się usadzić dzieci tak, żeby te, które się nie lubią siedziały w różnych autokarach?” — zamyślił się pan Romek. Spodobało mu się to, więc zamyślił się jeszcze raz. Takie usadzenie byłoby *absolutnie bezpieczne*. Szkoła dysponowała dwoma autokarami; gdyby takie rozdzielanie było możliwe, pan Romek byłby niepotrzebny i nie musiałby z nimi jechać.

„A jeśli nie istnieje absolutnie bezpieczne usadzenie dzieci? Wtedy *być może* mógłbym wybrać pewną parę nie lubiących się dzieci, wsadzić je do jednego autokaru, usiąść między nimi i przynajmniej ta para byłaby *zneutralizowana*” — stwierdził pan Romek. — „Pozostałe dzieci musiałbym jednak poumieszczać tak, żeby dla każdej innej nie lubiącej się pary jedno dziecko siedziało w jednym autokarze a drugie w drugim.” Takie usadzenie byłoby *bezpieczne*.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 500$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym wierszu wejścia znajdują się dwie nieujemne liczby całkowite  $n$  i  $m$ , oddzielone pojedynczą spacją, będące odpowiednio liczbą dzieci i liczbą par nie lubiących się dzieci. Dzieci numerowane są liczbami od 1 do  $n$ . W każdym z kolejnych  $m$  wierszy znajduje się para różnych liczb  $a_i$  i  $b_i$  oznaczająca, że dzieci  $a_i$  i  $b_i$  nie lubią się nawzajem.

## Ograniczenia danych

**Basic (a):**  $n \leq 3000$ ,  $m \leq 10^6$ . W całym pliku wejściowym jest co najwyżej 20 testów, w których  $n > 300$ .

**Professional (A):**  $n \leq 10^6$ ,  $m \leq 2 \cdot 10^6$ . W całym pliku wejściowym jest co najwyżej 5 testów, w których  $n + m > 30000$ .

## Wynik dla jednego zestawu

Jeśli pan Romek jest niepotrzebny, tj. istnieje absolutnie bezpieczne usadzenie dzieci, w pierwszym i jedynym wierszu wyniku należy wypisać słowo NIE. W przeciwnym przypadku w pierwszym i jedynym wierszu wyniku należy wypisać liczbę całkowitą z zakresu  $[0, m]$ , będącą liczbą wyborów takiej pary dzieci, że po jej neutralizacji przez pana Romka pozostałe dzieci można rozsadzić w bezpieczny sposób pomiędzy autokary.

## Przykład

Wejście	Wyjście
6	NIE
4 4	1
1 2	0
2 3	7
3 4	0
4 1	3
4 5	
1 2	
2 3	
3 4	
4 1	
1 3	
4 6	
1 2	
2 3	
3 4	
4 1	
1 3	
2 4	
7 7	
1 2	
2 3	
3 4	
4 5	
5 6	
6 7	
7 1	
6 6	
1 2	
2 3	
3 1	
4 5	
5 6	
6 4	
6 6	
1 2	
2 3	
3 1	
1 4	
2 5	
3 6	

## Zadanie B: Paspporty

Dezyderata, znana fałszerka dokumentów, westchnęła ciężko. Odkąd rząd postanowił pomóc obywatelom i wydał im nowe dowody, miała pełne ręce roboty. Kraj podzielono na dziesięć (nowych) województw ponumerowanych od 1 do 10, ale żeby ograniczyć ruch ludności, do województwa  $i$  można wjechać tylko, jeśli posiada się dowód osobisty, którego numer jest liczbą podzielną przez  $i$ .

Wszyscy mogą zatem wjechać do województwa warszawskiego, które otrzymało numer 1, ale akurat tym, nie wiedząc czemu, nie było zainteresowanych wielu klientów. Okazało się również, że numer na dowodzie bardzo trudno jest podrobić: każda cyfra rozpada się przy jakiegokolwiek próbie manipulacji. Natomiast dość szybko odkryto, że kolejność cyfr można dość łatwo zmieniać. Zarówno oryginalne jak i te podrobione przez Dezyderatę numery dowodów mogą zawierać zera wiodące.

Klienci Dezyderaty życzą sobie dodatkowo, żeby numer na ich dowodzie był najmniejszym możliwym do uzyskania, umożliwiającym wjazd do odpowiedniego województwa.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 100$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby całkowite dodatnie  $n$  i  $k$  oddzielone pojedynczą spacją, gdzie  $n$  jest numerem dowodu a  $k$  numerem województwa. Liczba  $n$  może zawierać zera wiodące.

### Ograniczenia danych

**Wspólne:**  $n \leq 10^{100\,000}$ .

**Basic (b):**  $k \leq 6$ .

**Professional (B):**  $k \leq 10$ .

### Wynik dla jednego zestawu

Twój program ma sprawdzić, czy po ewentualnej zamianie kolejności cyfr na dowodzie można będzie wjechać do województwa  $k$ . W przypadku negatywnym w pierwszym i jedynym wierszu wyjścia należy wypisać słowo NIE.

W przypadku pozytywnym w pierwszym i jedynym wierszu wyjścia należy wypisać utworzony w ten sposób numer dowodu. Wymagamy dodatkowo, żeby utworzony numer był najmniejszym spełniającym wymagania. Numer może zawierać zera wiodące.

### Przykład

Wejście	Wyjście
2	005
050 5	00000000035
50000000003 7	



## Zadanie C: Kalejdoskop

Pan Marian opatentował ostatnio nowoczesną konstrukcję kalejdoskopu. Zamiast szkielek i zwierciadeł głównym elementem takiego kalejdoskopu jest kwadratowy kawałek papieru. Na nim narysowano szachownicę  $n \times n$ , której każde pole jest pokolorowane na jeden z  $k$  kolorów. Po potrząśnięciu szachownica przesuwa się i przez wzornik widać pewien spójny fragment  $2 \times 2$  pola. Górny róg szachownicy jest skleiony z dolnym a prawy z lewym. Nie pytajcie, jak takie jednoczesne sklejenie jest możliwe — to jest właśnie ta część konstrukcji objęta patentem. I nie próbujcie tego w domu.

Dokładniej mówiąc, wiersze i kolumny szachownicy ponumerowane są od 0 do  $n - 1$ . Potrząsając kalejdoskopem, sprawiamy, że losowane są liczby  $a, b \in \{0, \dots, n - 1\}$  a przez wzornik widoczne są pola o następujących współrzędnych:

$(a, b)$	$((a + 1) \bmod n, b)$
$(a, (b + 1) \bmod n)$	$((a + 1) \bmod n, (b + 1) \bmod n)$

(Kalejdoskop ma wbudowany żyroskop i nie można nim obracać, licząc na to, że szachownica obróci się razem z kalejdoskopem).

Pan Marian chciałby teraz wyprodukować taką kwadratową szachownicę, żeby dla ustalonej liczby kolorów  $k$  można było zobaczyć (trzęsąc kalejdoskopem odpowiednio długo) dowolne wymarzone pokolorowanie czterech pól  $k$  kolorami. Szachownicę  $n \times n$  mającą tę własność nazywamy *kalejdoskopową*. Marian jest leniwym rysownikiem i chciałby, żeby  $n$  było jak najmniejsze. Łatwo zauważyć, że wymarzonych kolorowań czterech pól jest  $k^4$ , a zatem  $n$  musi być równe co najmniej  $k^2$ . Pan Marian dowiedział się, że radzieccy naukowcy udowodnili, że  $n = k^2$  wystarcza, ale niestety dostęp do pełnej wersji artykułu był płatny i pan Marian się nie skusił.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 20$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba pierwsza  $k$  będąca liczbą kolorów.

### Ograniczenia danych

**Basic (c):**  $k \leq 4$  i  $k$  jest liczbą pierwszą.

**Professional (C):**  $k \leq 40$  i  $k$  jest liczbą pierwszą.

### Wynik dla jednego zestawu

Należy wypisać kalejdoskopową szachownicę  $n \times n$ , gdzie  $n = k^2$ , tj. wypisać  $n$  wierszy, każdy zawierający  $n$  liczb oddzielonych pojedynczymi spacjami. Znajdująca się w wierszu  $i$ ,  $j$ -ta liczba powinna być liczbą całkowitą z zakresu  $[1, k]$  i oznaczać kolor pola szachownicy o współrzędnych  $(i - 1, j - 1)$ .

### Przykład

Wejście	Wyjście
1	1 2 1 1
2	1 2 2 2
	2 2 2 1
	1 1 2 1



## Zadanie D: Piaskownice

Pewnego dnia Cheopsowi znudziło się dotychczasowe życie, postanowił więc wyjść z piramidy i pobawić się. W pobliżu piramidy były piaskownice, ale w żadnej z nich nie było piasku! Nieco dalej znajdowały się kopalnie, a jak mu doniesiono, w każdej z nich znajdował się metr sześcienny piasku. „To nawet pasuje” — wykrzyknął Cheops — „każda piaskownica pomieści przecież jeden metr sześcienny piasku.” Druga część zdania była faktycznie prawdą, ale nie oznaczało to, że coś tu pasuje: liczba kopalni mogła być różna od liczby piaskownic, a i niekoniecznie wszystkie kopalnie były połączone ze wszystkimi piaskownicami drogami transportowymi. (Kopalnia może przesłać dowolną posiadaną część lub całość piasku do tych piaskownic, które są z nią połączone bezpośrednio drogami).

Cheops postanowił samodzielnie zadbać o dobrą zabawę i poprosił panią mamę Cheopsa o kieszonkowe. Cheops jest faraonem z natury zarówno zachłannym jak i skąpym, więc chciałby zamówić sumarycznie tyle piasku, ile tylko możliwe (uwzględniając dostępność piasku w kopalniach i pojemność piaskownic), ale zapłacić za to jak najmniej. Dla każdej kopalni określony jest przelicznik  $c$  (który może być różny dla każdej kopalni), oznaczający że kopalnia życzy sobie dostać  $c \cdot x^2$  kilogramów złota za sprzedaż  $x \text{ m}^3$  piasku.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 20$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite dodatnie  $k$  i  $p$  będące odpowiednio liczbą kopalni i liczbą piaskownic. W drugim wierszu znajduje się  $k$  nieujemnych liczb całkowitych  $c_1, c_2, \dots, c_k$  będących przelicznikami kolejnych kopalni. W kolejnych  $k$  wierszach znajduje się opis dróg. Dokładniej mówiąc, dla  $i \in [1, k]$ , w wierszu  $i + 2$  znajduje się  $p$  liczb oddzielonych pojedynczymi spacjami:  $j$ -ta z tych liczb jest równa 1, jeśli kopalnia  $i$  jest bezpośrednio połączona drogą z piaskownicą  $j$ , i jest równa 0, jeśli brak takiego połączenia.

## Ograniczenia danych

**Wspólne:**  $k \leq 100, p \leq 100$ .

**Basic (d):**  $c_i = 1$  dla każdej kopalni  $i$ .

**Professional (D):**  $c_i$  jest liczbą całkowitą z przedziału  $[0, 100]$  dla każdej kopalni  $i$ .

## Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę nieujemną oznaczającą minimalną liczbę kilogramów złota, jaką Cheops musi zapłacić za transport maksymalnej możliwej ilości piasku z kopalni do piaskownic. Twoja odpowiedź zostanie zaakceptowana, jeśli zostanie udzielona z błędem względnym lub bezwzględnym mniejszym niż  $10^{-5}$ .

## Przykład

Wejście	Wyjście
4	0.500000
2 1	2.000000
1 1	5.454545
1	1.333333
1	
2 1	
4 4	
1	
1	
3 2	
5 3 5	
1 0	
1 1	
0 1	
4 2	
1 1 1 1	
1 0	
1 0	
1 0	
0 1	





## Zadanie E: Emerytury

— Czesław! Budżet mi się znowu nie domyka!  
— To weź sobie kasę z OFE.  
— Już wziąłem. Ale pomimo to nie ma z czego wyżywić 10 milionów jeńców, to znaczy, tych, no, emerytów.  
— Oj tam, oj tam, to wypłać emerytury tylko niektórym, na przykład ponumeruj ich i wypłać tylko, jak czyjś numer jest liczbą złożoną...  
— Chyba na głowę upadłeś, przecież prawie każda liczba jest złożona.  
— ... no to liczbą będącą iloczynem liczb Fibonacciego. Niekoniecznie różnych, być może tylko jednej.  
— Czesław, ty jak zrobisz ustawę to nie ma nikogo we wsi! I jeszcze na waciki będziemy mieli!  
— Ciesz się, że to nie jeńcy, bo byśmy mieli na karku smutnych panów od konwencji genewskiej.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 50$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym i jedynym wierszu wejścia znajduje się liczba całkowita dodatnia  $n$ , będąca liczbą emerytów. Emeryci są numerowani liczbami od 1 do  $n$ .

### Ograniczenia danych

**Basic (e):**  $n \leq 10^{11}$ .

**Professional (E):**  $n \leq 10^{17}$ .

### Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać liczbę emerytów, którym zostaną wypłacone pieniądze. (Warto pamiętać, że liczby Fibonacciego można zdefiniować rekurencyjnie jako  $F_0 = F_1 = 1$  i  $F_k = F_{k-1} + F_{k-2}$  dla  $k \geq 2$ ).

## Przykład

Wejście	Wyjście
6	1
1	9
10	9
11	10
12	48
100	198
1000	



## Zadanie F: Kołderka

Jaś hoduje w pudełku dżdżownice. Dżdżownice są dobrze wychowane i dlatego leżą nieruchomo w pudełku, tak jak zostały ułożone przez Jasia: poziomo, z częściami głowowymi skierowanymi w lewą stronę. Wszystkie dżdżownice mają tę samą długość i składają się z  $k$  segmentów. Wszystkie segmenty dżdżownic są takiej samej wielkości i różnią się tylko kolorem. Dżdżownice da się odróżnić, tj. każde dwie różnią się kolorem choć jednego segmentu.

Jaś do dżdżownic ma stosunek mocno ambiwalentny: z jednej strony je kocha, z drugiej zasadniczo się ich brzydzi. Uszył im więc kołderkę (żeby im nie było zimno i żeby nie musiał ich oglądać w całej okazałości), która ma wysokość taką jak pudełko i pewną szerokość  $\ell \leq k$  liczoną w dżdżowniczych segmentach. Kołderkę (szeroką taką) można przesuwac w lewo lub prawo, ale nie można jej obracać. Dokładniej mówiąc, po ułożeniu kołderki o szerokości  $\ell$  wystaje spod niej  $\ell_1$  pierwszych segmentów każdej dżdżownicy i  $\ell_2$  ostatnich segmentów każdej dżdżownicy, gdzie  $\ell_1 + \ell + \ell_2 = k$ .

Jaś chciałby, żeby kołderka była jak najszersza. Pewnie zastanawiacie się, dlaczego kołderka nie ma po prostu szerokości  $k$  i nie przykrywa całkowicie każdej dżdżownicy. Otóż po wystających częściach Jaś chce móc zidentyfikować każdą z dżdżownic. Chce mianowicie, żeby dla dowolnych dwóch dżdżownic istniał wystający poza kołderkę segment, taki że kolor tego segmentu jest różny dla obu tych dżdżownic.

### Opis testów

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 20$  oznaczająca liczbę zestawów testowych, które są opisane w kolejnych wierszach. Każdy z zestawów jest zgodny ze specyfikacją podaną w części *Jeden zestaw danych*. Program powinien wypisywać wyniki na *wyjście standardowe*. Wyniki dla poszczególnych zestawów powinny być zgodne ze specyfikacją opisaną w części *Wynik dla jednego zestawu* i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite dodatnie  $n$  i  $k$  oddzielone pojedynczą spacją, gdzie  $n \geq 2$  jest liczbą dżdżownic. W każdym z kolejnych  $n$  wierszy znajduje się opis jednej dżdżownicy, będący ciągiem  $k$  małych liter alfabetu łacińskiego nieoddzielonych odstępami. Każda litera odpowiada jednemu kolorowi. Powyższy ciąg oznacza zatem kolory kolejnych segmentów od części głowowej do części ogonowej dżdżownicy.

## Ograniczenia danych

**Basic (f):**  $n, k \leq 200$ .

**Professional (F):**  $n \cdot k \leq 10^6$ .

## Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyjścia należy wypisać nieujemną liczbę całkowitą  $\ell$  będącą maksymalną szerokością kołderki, która po odpowiednim ułożeniu pozwoli na identyfikację wszystkich dżdżownic. W przypadku, gdy dowolne przykrycie kołderką o niezerowej szerokości uniemożliwia identyfikację, należy wypisać liczbę 0.

## Przykład

Wejście	Wyjście
4	1
3 3	1
aaa	4
abb	0
baa	
4 3	
aba	
aab	
baa	
bab	
4 5	
aaaaa	
aaaab	
aaaac	
aaaad	
3 2	
ba	
aa	
ab	

