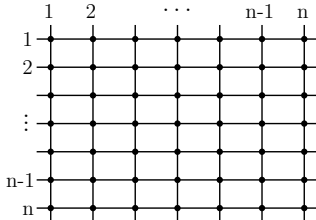


## Zadanie A: Projekt Manhattan

Manhattan składa się z  $n$  równoległych ulic poziomych (ponumerowanych od 1 do  $n$ ) i  $n$  równoległych ulic pionowych (również ponumerowanych od 1 do  $n$ ). Ulice są idealnie proste, a ulice poziome są prostopadłe do ulic pionowych, tak jak na poniższym rysunku.



Pewna stacja telewizyjna postanowiła prezentować wszystkie zdarzenia na ulicach Manhattanu. Człowiek stojący na skrzyżowaniu drogi poziomej  $i$  z drogą pionową  $j$  widzi, co dzieje się na wszystkich skrzyżowaniach tych dwóch ulic z pozostałymi. Dla każdego skrzyżowania ustalono cenę, za którą można znaleźć blogera, który będzie skłonny na nim pracować (i przekazywać z obu widzianych przez siebie ulic pozbawione znaczenia wiadomości wstrząsające opinią publiczną). Okazało się, że dla niektórych skrzyżowań taka cena jest ujemna (niektórzy blogerzy dla lansu gotowi są zapłacić swojemu pracodawcy). Stacja telewizyjna zastanawia się, jak kontrolować wszystkie skrzyżowania i jednocześnie zminimalizować sumaryczne opłaty dla blogerów.

### Test

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 znajduje się liczba całkowita dodatnia  $n$ . W kolejnych  $n$  wierszach znajduje się opis kolejnych poziomych ulic. Opis jednej poziomej ulicy zawiera  $n$  oddzielonych pojedynczymi spacjami liczb, będących cenami wynajmu blogera na kolejnych skrzyżowaniach tej ulicy.

### Ograniczenia danych

**Basic (a):**  $n \leq 8$ . Wszystkie ceny są z zakresu  $[1, 10^6]$ .

**Professional (A):**  $n \leq 500$ . Wszystkie ceny są z zakresu  $[-10^6, 10^6]$ .

### Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać jedną liczbę całkowitą, będącą minimalną ceną, którą stacja telewizyjna musi zapłacić, żeby zagwarantować sobie dostęp do wszystkich ulic.

### Przykład

Wejście	Wyjście
1	6
4	
8 4 2 9	
7 1 8 3	
8 1 4 3	
3 2 8 7	

## Zadanie B: Liczba i żaba

Za siedmioma górami, za siedmioma lasami żyła sobie piękna, niezależna i pewna siebie liczba  $n$ , która wszędzie miała daleko. Pewnego dnia spotkała żabę szerokoustną.

— Cześć, jestem żaba — powiedziała żaba.

— A ja jestem bocian i szukam żab szerokoustnych — odpowiedziała niezbyt zgodnie z prawdą liczba  $n$ .

— Słuchaj, pocałuj mnie, a zamienię się w księcia z bajki, zamieszkamy razem z moją matką i będziesz mi gotować, rodzić dzieci i prać moje brudne skarpetki.

— No jakoś nie sędzę — odpowiedziała liczba  $n$ , choć na usta cisnęły jej się słowa wujka Staszka, mistrza ciętej riposty.

— To może się chociaż poprawnie przedstawić?

— Że niby jak?

— No na przykład jako sumę dwóch liczb pierwszych. Założę się, że nie dasz rady.

## Przykład

Wejście	Wyjście
4	NIE
1	61 7
68	101 2
103	1000000007 1000000009
2000000016	

## Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba  $Z \leq 200$  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  $n$ . Taka z bajki.

## Ograniczenia danych

**Basic (b):**  $n \leq 10^6$ .

**Professional (B):**  $n \leq 10^{12}$ .

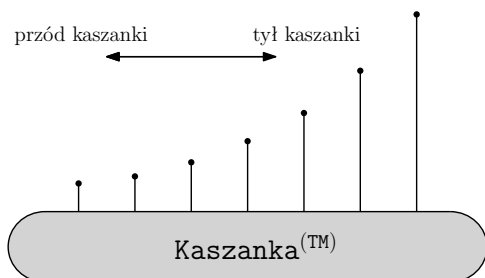
## Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać dwie liczby pierwsze  $x$  i  $y$  rozdzielone pojedynczą spacją takie, że  $x + y = n$ , lub napis NIE, jeśli takie liczby nie istnieją.

## Zadanie C: Jesz (zupę)

W tegorocznym konkursie układania mięsnych jeży motywem przewodnim jest mięso z muchami. Jaś postanowił wygrać konkurs. W tym celu zakupił długą kaszankę i zabrał ją do domu. Następnie pożyczył od Kuby  $n$  patyczków różnych długości, ponumerowanych od 1 do  $n$ . Patyczki te zamierza wbić w kaszankę (jak jeź, to jeź!) w jednym szeregu, w równych odstępach. Każdy patyczek wbijany jest na taką samą, zaniedbywalnie małą głębokość.

Okazało się, że na kaszance jest miejsce tylko na  $k$  z nich, jednak — żeby kaszankę nie gardził człowiek — muszą one być w oryginalnej kolejności (zgodnie z numeracją). Jaś złapał również odpowiednią liczbę żywych much i przykleił po jednej na górze każdego patyczka. Żeby jeź zachwyił jurorów, musi latać, a w tym celu spełniać dwa dodatkowe wymagania techniczne. Kaszanka jaka jest, każdy widzi (na poniższym rysunku), ma przód i tył i lata równoległe do ziemi. Muchy na rysunku zaznaczone są kropkami.



Dowolna mucha patrząca do przodu nie może mieć zasłoniętego widoku przez poprzednie patyczki, tj. każdy kolejny patyczek musi mieć większą długość niż poprzedni. Dla większego bezpieczeństwa każda mucha musi widzieć wszystkie inne (muchy mogą patrzeć jednocześnie do przodu i do tyłu), tzn. prosta łącząca czubki dowolnych dwóch patyczków musi leżeć ściśle ponad patyczkami pośrednimi.<sup>1</sup> Muchy są małe i nie przeszkadzają w widzeniu, ani nie dodają wysokości patyczkom. Jaś zastanawia się, czy da się wybrać odpowiednie  $k$  patyczków. Kaszanka musi fruwać!

### Test

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*

<sup>1</sup> „Ściśle ponad” oznacza, że musi leżeć ponad patyczkami pośrednimi i nie może ich dotykać.

i należy je wypisać w takiej kolejności, w jakiej zestawy występują na wejściu.

### Jeden zestaw danych

W pierwszym wierszu wejścia znajdują się liczby całkowite dodatnie  $n$  i  $k$  oddzielone pojedynczą spacją. W drugim i ostatnim wierszu wejścia znajduje się  $n$  liczb całkowitych, oddzielonych pojedynczymi spacjami, będących długościami kolejnych patyczków.

### Ograniczenia danych

**Wspólne:**  $k \leq n \leq 20\,000$ . Długość każdego z patyczków jest liczbą całkowitą z zakresu  $[1, 7 \cdot 10^8]$ .

**Basic (c):**  $k \leq 3$ .

**Professional (C):**  $k \leq 100$ .

### Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać  $k$  liczb całkowitych z zakresu  $[1, n]$  będących numerami patyczków użytych do konstrukcji latającego jeża. Liczby powinny być oddzielone pojedynczymi spacjami. Pamiętaj, że patyczki muszą być w oryginalnej kolejności. Jeśli z danych patyczków nie da się skonstruować latającego jeża, w pierwszym i jedynym wierszu wyjścia należy wypisać słowo NIE.

### Przykład

Wejście	Wyjście
7	4 5 6
6 3	2 3 5
4 6 5 1 2 4	NIE
6 3	NIE
1 10 11 12 13 14	1 2 4 7
3 3	3 4 7 8
1 1 1	NIE
7 5	
5 1 7 8 9 3 4	
10 4	
3 5 10 8 9 1 12 13 8	
9	
12 4	
3 4 1 2 2 3 4 8 3 5 3	
4	
12 5	
3 4 1 2 2 3 4 8 3 5 3	
4	

## Zadanie D: Przekupstwo

W pewnym turnieju w szranki stanęło (a właściwie w szeregu ustawiło się)  $n$  osób. Okazało się że wszyscy są przedstawicielami jednego z trzech Niezwykłe Popularnych Zawodów: każdy jest albo naukowcem, albo parasolnikiem, albo kurtyzana. Turniej odbywa się w  $n - 1$  rundach. W każdej rundzie sędzia wybiera dwie stojące obok siebie osoby i każe im ze sobą walczyć; osoba, która przegrała, zostaje usunięta z szeregu. Zwycięzcą zostaje ostatnia nieusunięta osoba.

Oczywiście w walce wszystkie chwytów są dozwolone. Jednak okazuje się, że jeśli walczą przedstawiciele dwóch różnych zawodów to wynik jest z góry przesądzony: naukowiec zagaduje parasolnika na śmierć, parasolnik wybijają kurtyzanie oko parasolem, a kurtyzana robi z naukowcem co chce (względnie to, co umie najlepiej). Jeśli natomiast walczą ze sobą przedstawiciele takich samych zawodów, to nie da się przewidzieć wyniku (choć dla spragnionych wrażeń widzów wynik ma wtedy drugorzędne znaczenie). Warto podkreślić, że w żadnym przypadku sędzia nie ma żadnego wpływu na wynik takiej pojedynczej walki.

Jednak to, kto zostanie na placu boju, bardzo zależy od kolejności rozgrywek. Dlatego też każdy z zawodników zastanawia się, czy ma sens przekupić sędziego, tj. czy sędzia może tak wybierać osoby do kolejnych rozgrywek, żeby *zagwarantować* zwycięstwo danemu zawodnikowi.

### Test

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 wierszu wejścia znajduje się liczba całkowita dodatnia  $n$ , będąca liczbą zawodników w szeregu. W drugim wierszu wejścia znajduje się opis zawodników w szeregu będący ciągiem (nieoddzielonych)  $n$  znaków. Każdy znak należy do zbioru  $\{N, P, K\}$  i oznacza, że na danym miejscu stoi odpowiednio naukowiec, parasolnik lub kurtyzana.

### Ograniczenia danych

**Basic (d):**  $n \leq 300$ .

**Professional (D):**  $n \leq 10^6$

### Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać ciąg  $n$  cyfr (bez odstępów). Cyfra 1 oznacza, że zawodnikowi na danej pozycji można zagwarantować zwycięstwo przekupując sędziego, zaś cyfra 0 oznacza, że jest to niemożliwe.

### Przykład

Wejście	Wyjście
3	00
2	101
PP	100101
3	
NKP	
6	
PPKNPK	

## Zadanie E: Krzeselka

Po krwawej jatce, którą przedstawiciele Niezwykłe Popularnych Zawodów (nauczyciele, parasolnicy i kurtyzany) urządzili w Zadaniu D, niedobitki postanowiły wspólnie zabiegać o fundusze strukturalne Unii Europejskiej, zgodnie z maksymą, że zarządzania grantem unijnym życzy się tylko najgorszemu wrogowi. Po złożeniu liczącego 1410 stron wniosku, sprawdzonego przez 69 urzędników i 44 recenzentów, Unia postanowiła przyznać dofinansowane na krzeselka, które miały stanąć przed każdą z  $n$  budek z piwem w wiosce (po jednym krzeselku na budkę). Oczywiście każde krzeselko zostało zasłonięte wielką planszą z informacją o źródle finansowania.

Budki w wiosce połączone są miedzami, a każda miedza łączy dwie budki. Miedź jest o jedną mniej niż budek, ale mimo to z każdej budki da się dojść miedzami do każdej innej (na mieście ludzie mówili o cudzie!). Dwie budki nazywamy *sąsiednimi*, jeśli połączone są miedzą.

Okazało się, że liczba budek z piwem w wiosce jest liczbą podzielną przez trzy i dzięki temu każdej grupie zawodowej przypadła taka sama liczba krzeselek. Jednak gdy postanowiono się nimi równo podzielić, okazało się, że Unia Europejska, powołując się na ustanowioną właśnie Dyrektywę o Trzeźwości w Pracy i Prawo o Zakazie Nieuczciwej Konkurencji dla Niezwykłe Popularnych Zawodów, zabroniła przedstawicielom tej samej grupy siedzieć na sąsiednich krzeselkach. Pomóż im podzielić się miejscami, żeby nie narzekali, że w tym kraju trzeba tak pić.

### Test

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 znajduje się liczba całkowita dodatnia  $n$ , podzielna przez 3 i będąca liczbą budek. Wszystkie budki numerowane są od 1 do  $n$ . W kolejnych  $n - 1$  wierszach znajduje się opis miedz: w wierszu  $i \in \{2, \dots, n\}$  znajduje się liczba całkowita dodatnia  $p_i \leq i$  oznaczająca, że istnieje miedza łącząca budki  $i$  oraz  $p_i$ .

### Ograniczenia danych

**Wspólne:**  $n \leq 500\,000$ .

**Basic (e):** Każda budka sąsiaduje z co najwyżej trzema innymi.

**Professional (E):** Każda budka sąsiaduje z co najwyżej czterema innymi.

### Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać ciąg  $n$  liter ze zbioru  $\{N, K, P\}$  (bez odstępów). Stojąca na miejscu  $i$  litera  $N$  oznacza, że na krzeselku przy budce  $i$  zasiadzie naukowiec,  $K$  — kurtyzana, zaś  $P$  — parasolnik. Pamiętaj o warunkach narzuconych przez unijną biurokrację! Jeśli nie da się ich spełnić, w pierwszym i jedynym wierszu wyniku należy wypisać słowo NIE.

### Przykład

Wejście	Wyjście
2	NPKPN
6	NKP
1	
1	
1	
2	
2	
3	
1	
1	

## Zadanie F: Dyrektor

Po przeprowadzce do nowego królestwa pewien dyrektor postanowił dobrze zabezpieczyć swój samochód. A było co zabezpieczać, bo Maybach do tanich nie należał. Nawiązał więc znajomość z  $m$  potężnymi magami (przez myśl przeszło mu odnowić swoje konszachty z księżmi, ale — zupełnie jak w RPG — magowie dysponowali lepszymi czarami obronnymi) i wynajął  $n = 3 \cdot m$  wysokich wież.

Postanowił, że każdemu magowi przydzieli 3 wieże. Trójkąt, którego wierzchołkami są wieże należące do konkretnego maga, jest kontrolowany przez tego maga. Oczywiście każda wieża może być zajęta tylko przez jednego maga. Dyrektor chciałby, żeby miejsce, w którym zaparkuje Maybacha, było kontrolowane przez wszystkich magów. W tym celu musi przydzielić każdemu z nich odpowiednie wieże. Wróżka Zębuszka przepowiedziała dyrektorowi, że da się to zrobić. Niestety za dalszą ekspertyzę kazała sobie słono zapłacić, zatem dyrektor zwrócił się do Ciebie z prośbą o pomoc.

### Test

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 znajduje się liczba całkowita dodatnia  $n$ , podzielna przez 3 i będąca liczbą wież. W każdym z kolejnych  $n$  wierszy znajduje się opis jednej wieży, będący parą liczb całkowitych oddzielonych pojedynczą spacją, będących współrzędnymi wieży. Wszystkie wieże są parami różne i żadne trzy wieże nie leżą na jednej prostej.

### Ograniczenia danych

**Wspólne:**  $n \leq 200$ . Każda ze współrzędnych jest z zakresu  $[-10^5, 10^5]$ .

**Basic (f):** Wszystkie wieże są wierzchołkami pewnego wielokąta wypukłego.

**Professional (F):** Pozycje wież są dowolne.

### Wynik dla jednego zestawu

W  $n/3$  kolejnych wierszach wyjścia należy wypisać przypisanie wież do poszczególnych magów. W wierszu  $i$  należy wypisać trzy liczby całkowite z zakresu  $[1, n]$ , oddzielone pojedynczymi spacjami, będące numerami wież przydzielonych magowi  $i$ .

### Przykład

Wejście	Wyjście
1	5 1 3
6	4 6 2
-10 0	
10 0	
0 -10	
0 10	
10 10	
-10 -10	