Zadanie A: Proste

Eustachy dostał na urodziny n prostych, z których żadne trzy nie przecinają się w jednym punkcie. Z braku lepszych zajęć (i lepszych prezentów) zastanawia się teraz nad następującą kwestią: jaka jest najmniejsza możliwa liczba punktów, które można wybrać tak, aby na każdej prostej leżał przynajmniej jeden z wybranych punktów? Wybrane punkty mogą mieć dowolne współrzędne.

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 100$ 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 wierszu wejścia znajduje się liczba całkowita dodatnia n. W kolejnych n wierszach znajduje się opis kolejnych prostych. Opis i-tej prostej składa się z dwóch liczb całkowitych a_i oraz b_i oddzielonych pojedynczą spacją, które reprezentują prostą $y = a_i \cdot x + b_i$. Podane na wejściu proste nie powtarzają się, a ponadto żadne trzy z nich nie przecinają się w jednym punkcie.

Ograniczenia danych

Wspólne: $a_i, b_i \in [-10^9, 10^9]$ dla każdego $i = 1, 2, \ldots, n$.

Basic (a): $n \leqslant 3$.

Professional (A): $n \leqslant 50000$.

Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać najmniejszą możliwą liczbę punktów, które można wybrać tak, aby na każdej prostej leżał przynajmniej jeden z wybranych punktów.

Przykład

Wejście	Wyjście
3 3 1 10 1 20 1 30 3 0 1 1 2 2 4 1 -1 -1	3 2 1

Zadanie B: Nawiasy

Zenobiusz kupił w sklepie z prawidłowo ponawiasowanymi wyrażeniami napis długości n. Niestety, okazało się, że nazwa sklepu jest, delikatnie mówiąc, myląca, i być może napis wcale nie jest prawidłowo ponawiasowanym wyrażeniem! Zenobiuszowi nie pozostaje więc nic innego niż wyciąć niektóre znaki napisu tak, aby uzyskać prawidłowo ponawiasowane wyrażenie. Usunięcie i-tego znaku wiąże się ze spadkiem ogólnej wartości estetycznej całego napisu o c_i. Jak usunąć niektóre (być może żadne) znaki podanego napisu tak, aby otrzymać prawidłowo ponawiasowane wyrażenie, a jednocześnie zminimalizować całkowity koszt? Całkowity koszt jest rozumiany jako suma spadków ogólnej wartości estetycznej wszystkich usuniętych znaków.

Prawidłowo ponawiasowane wyrażenie definujemy w następujący rekurencyjny sposób.

1. pusty napis jest prawidłowo ponawiasowanym wyrażeniem,
2. jeśli S jest prawidłowo ponawiasowanym wyrażeniem, to takim wyrażeniem jest także (S),
3. jeśli S i T są prawidłowo ponawiasowanymi wyrażeniami, to takim wyrażeniem jest także ich konkatenacja ST.

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 10$ 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 drugim wierszu znajduje się napis złożony z n znaków, z których każdy to nawias otwierający lub nawias zamykajacy, czyli (lub). W trzecim wierszu znajduje się n liczb całkowitych $c_1, c_2, \ldots, c_n$ oddzielonych pojedynczymi spacjami, które oznaczają spadki ogólnej wartości estetycznej po usunięciu kolejnych znaków podanego wcześniej napisu.

Ograniczenia danych

Wspólne: $n \leqslant 10^6$.

Basic (b): $c_i = 1$ dla każdego $i = 1, 2, \ldots, n$.

Professional (B): $c_i \in [1, 10^9]$ dla każdego $i = 1, 2, \ldots, n$.

Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać najmniejszy możliwy całkowity koszt usunięcia niektórych (lub być może żadnego) znaków podanego napisu tak, aby uzyskać prawidłowo ponawiasowane wyrażenie.

Przykład

Wejście	Wyjście
2 5 ((() 1 1 1 1 1 6 ())()(1 1 1 1 1 1	1 2

Zadanie C: Promienie

W pewnym odległym kraju o niewymawialnej nazwie, która zaczyna się na literę B, każde miasto składa się n wieżowców ustawionych w szereg. Wysokości wieżowców zawsze tworzą permutację liczb $1, 2, \ldots, n$. To, jak dokładnie wygląda ta permutacja, nie jest szczególnie ważne dla mieszkańców. Kluczowe ze względów religijno-kulturowych są dla nich jednak *promienie* kolejnych wieżowców. Promień i-tego wieżowca, oznaczany przez r_i, jest największą liczbą całkowitą o takiej własności, że wieżowce o numerach $i - r_i, i - r_i + 1, \ldots, i + r_i - 1, i + r_i$ są wszystkie niższe niż ten o numerze i (wymagamy także, żeby te wszystkie wieżowce istniały, czyli $i - r_i \geqslant 1$ oraz $i + r_i \leqslant n$).

Mieszkańcy nie znają wysokości wieżowców, z których składa się ich miasto, wiedzą tylko, że tworzą one permutację liczb $1, 2, \ldots, n$. Promienie wszystkich wieżowców $r_1, r_2, \ldots, r_n$ są jednak doskonale znane każdemu z nich. Co bardziej dociekliwi z młodszych mieszkańców zastanawiają się czasami, ile jest permutacji, które odpowiadają tym znanym promieniom. Czy potrafisz im pomóc?

Uwaga: być może mieszkańcy zostali wprowadzeni w błąd i znane przez nich promienie nie są prawidłowe, to znaczy nie odpowiada im żadna permutacja.

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 100$ 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 wierszu wejścia znajduje się liczba całkowita dodatnia n. W drugim wierszu znajduje się n liczb całkowitych $r_1, r_2, \ldots, r_n$ oddzielonych pojedynczymi spacjami, które oznaczają promienie kolejnych wieżowców.

Ograniczenia danych

Wspólne: $r_i \in [0, 10^9]$ dla każdego $i = 1, 2, \ldots, n$.

Basic (c): $n \leqslant 1000$. Istnieje przynajmniej jedna permutacja odpowiadająca podanym promieniom.

Professional (C): $n \leqslant 5000$.

Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać liczbę permutacji, które odpowiadają podanym promieniom, modulo $10^9 + 7$. Jeśli nie ma żadnej takiej permutacji, zamiast liczby należy wypisać słowo NIE.

Przykład

Wejście	Wyjście
4 6 0 1 0 0 1 0 5 0 0 2 0 0 3 0 1 0 7 0 1 0 3 0 0 0	80 24 2 160

Zadanie D: Gra

Ambroży spędza większość swojego wolnego czasu grając w wymyśloną przez siebie jednoosobową grę w zabieranie par krawędzi. Do gry potrzebny jest graf nieskierowany, w którym każda krawędź ma przypisaną pewną wagę. Ambroży wykonuje zero lub więcej ruchów, z których każdy polega na wybraniu pary krawędzi, które mają wspólny koniec. Następnie wagi obydwu krawędzi są dodawane do wyniku, a same krawędzie zostają usunięte z grafu.

Choć pewnie nigdy by się do tego nie przyznał, Ambroży zwykle wybiera usuwane pary krawędzi bez szczególnego zastanowienia. Czasami zastanawia się jednak, jaki jest największy możliwy wynik, który można uzyskać dla danego grafu? Dodatkowo, chciałby wiedzieć jakie pary krawędzi należy kolejno usuwać, aby uzyskać taki wynik.

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 50$ 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 znajdują się liczba całkowita dodatnia n i liczba całkowita m oddzielone pojedynczą spacją, które oznaczają odpowiednio liczbę wierzchołków i krawędzi grafu. W kolejnych m wierszach znajdują się opisy kolejnych krawędzi. Opis i-tej krawędzi zawiera trzy liczby całkowite x, y, w oddzielone pojedynczymi spacjami, które oznaczają numery wierzchołków połączone tą krawędzią oraz jej waga. Wierzchołki numerujemy liczbami naturalnymi od 1 do n.

Ograniczenia danych

Wspólne: $x_i, y_i \in [1, n]$, $x_i \neq y_i$ oraz $w_i \in [0, 10^9]$ dla każdego $i = 1, 2, \ldots, m$. Podane krawędzie nie powtarzają się.

Basic (d): $n, m \leqslant 1000$.

Professional (D): $n \leqslant 10^5$, $m \leqslant 10^6$.

Wynik dla jednego zestawu

W pierwszym wierszu wyniku należy wypisać jedną liczbę całkowitą, będącą największym możliwym wynikiem w opisanej grze. W drugim wierszu należy wypisać jedną liczbą całkowitą k, będącą liczbą par krawędzi, których kolejne zabieranie gwarantuje taki wynik. W kolejnych k wierszach należy wpisać te pary krawędzi. Krawędzie numerujemy liczbami naturalnymi od 1 do m zgodnie z kolejnością podania na wejściu.

Przykład

Wejście	Wyjście
4 4 3 1 2 20 2 3 10 3 4 30 3 3 1 2 1 2 3 2 3 1 3 5 4 1 2 1 1 3 10 1 4 100 1 5 1000 6 4 1 2 1000000 1 3 1000000 4 5 1000000 4 6 1000000	40 1 2 3 5 1 2 3 1111 2 1 2 3 4 4000000 2 1 2 3 4

Zadanie E: Autostrada

Tuż po uroczystym otwarciu pierwszej autostrady (nazwanej Autostradą) we Flatlandii okazało się, że całkiem zapomniano o zaplanowaniu jakichkolwiek dróg dojazdowych. Mało tego: cała autostrada jest ogrodzona wysokim murem, więc w tej chwili nikt nie może z niej korzystać! Jak łatwo się domyślić, niezbyt ucieszyło to mieszkańców pobliskich n wiosek. Ku ich zaskoczeniu, ministerstwo transportu przyznało się do swojego niedopatrzenia i w ekspresowym terminie wydało zgodę na utworzenie co najwyżej k wjazdów na nowo wybudowaną autostradę. Pozostaje jednak kwestia wybrania miejsc, w których należy je utworzyć. Mieszkańcy zgodnie stwierdzili, że chcieliby wybrać wjazdy tak, aby zminimalizować sumę odległości mieszkańców do autostrady, gdzie odległość mieszkańca do autostrady to odległość jego wioski do najbliższego wjazdu. Dziwnym zbiegiem okoliczności wszyscy mieszkańcy pracowali przez kilka lat jako taksówkarze w Nowym Yorku, w związku z tym przez odległość między punktem (x, y) a (x', y') rozumieją wartość $|x - x'| + |y - y'|$, gdzie $||$ oznacza wartość bezwzględną (czyli liczą odległość w metryce taksówkarza).

Autostrada jest prostą opisaną przez równanie $y = ax+b$. i-ta z wiosek znajduje się w punkcie (x_i, y_i) i mieszka w niej dokładnie w_i mieszkańców. Każdy z wjazdów na autostradę musi być punktem na prostej $y = ax+b$ (o niekoniecznie całkowitych współrzędnych). Jaka jest najmniejsza możliwa suma odległości mieszkańców od autostrady po utworzeniu co najwyżej k wjazdów?

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 40$ 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 a i b oddzielone pojedynczą spacją, które definiują autostradę $y = ax + b$. W drugim wierszu znajdują się dwie liczby całkowite dodatnie n i k oddzielone pojedynczą spacją, które oznaczają liczbę wiosek i maksymalną liczbę wjazdów. W kolejnych n wierszach znajdują się opisy kolejnych wiosek. Opis i-tej wioski składa się z trzech liczb całkowitych x_i, y_i i w_i oddzielonych pojedynczymi spacjami, gdzie (x_i, y_i) oznacza położenie wioski, a w_i liczbę jej mieszkańców.

Ograniczenia danych

Wspólne: : $a \in [-100, 100]$, $b \in [-10^9, 10^9]$, $x_i, y_i \in [-10^9, 10^9]$ oraz $w_i \in [1, 100]$ dla każdego $i = 1, 2, \ldots, n$.

Basic (e): $n, k \leqslant 100$.

Professional (E): $n \leqslant 1000$, $k \leqslant 10^9$.

Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać jedną liczbę rzeczywistą, będącą najmniejszą możliwą sumą odległości mieszkańców do autostrady po utworzeniu co najwyżej k wjazdów na podanej autostradzie. Za poprawną zostanie uznana odpowiedź, która różni się od wzorcowej o co najwyżej 0.01.

Przykład

<table><tr><th>Wejście</th><th>Wyjście</th></tr><tr><td><pre>4
0 0
3 1
-10 10 1
0 10 1
10 10 1
1 0
3 2
6 5 4
0 2 1
2 -1 1
0 4
6 3
-2 4 6
2 6 1
3 2 1
4 6 1
5 2 1
6 0 1
97 0
1 1
23 32 99</pre></td><td><pre>50.00
9.00
15.00
2244.34</pre></td></tr></table>

Zadanie F: Czworościan

Na prawie każdych zawodach uczestnicy głośno domagają się choć jednego zadania z geometrii, a najlepiej geometrii 3D. Uprzedzając takie żądania, w ostatnim zadaniu w tym zestawie należy policzyć objętość podanego czworościanu. Czworościan jest zadany przez współrzędne (x_i, y_i, z_i) kolejnych wierzchołków dla $i = 1, 2, 3, 4$.

Test

Program powinien czytać dane z *wejścia standardowego*. W pierwszym wierszu podana jest liczba $Z \leqslant 100$ 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

Wejście składa się z czterech linii. W i-tej z nich znajdują się trzy liczby całkowite x_i, y_i, z_i, które oznaczają współrzędne i-tego wierzchołka czworościanu.

Ograniczenia danych

Basic (f): $x_i, y_i \in [-1000, 1000]$ dla każdego $i = 1, 2, 3, 4$, $z_1 = z_2 = z_3 = 0$ oraz $z_4 \in [-1000, 1000]$.

Professional (F): $x_i, y_i, z_i \in [-1000, 1000]$ dla każdego $i = 1, 2, 3, 4$.

Wynik dla jednego zestawu

W pierwszym i jedynym wierszu wyniku należy wypisać jedną liczbę rzeczywistą, będącą objętością podanego czworościanu. Za poprawną zostanie uznana odpowiedź, której błąd względny lub bezwzględny nie przekracza 10^{-6}.

Przykład

Wejście	Wyjście
2 0 0 0 10 0 0 0 10 0 0 0 10 1 2 3 6 5 4 12 8 9 11 12 13	166.666667 16.666667