

# AISDI zadanie GRAF

## Proszę napisać program, który:

1. wczytać ze standardowego wejścia graf skierowany,
2. wyszukać w zadanym grafie wszystkie cykle proste składające się z parzystej liczby krawędzi oraz dla każdego z tych cykli policzyć sumę wag krawędzi,
3. wypisać na standardowe wyjście numery kolejnych wierzchołków każdego znalezionej cyklu prostego oraz wagę tego cyklu.

## Dodatkowe założenia:

1. krawędzie mogą być tylko jednokrotne – nie rozważamy przypadku multigrafu,
2. próba dodania kolejny raz wagi dla istniejącej już krawędzi powinna nadpisać dotychczasową wartość,
3. wagi mają wartości nieujemne,
4. dla wszystkich danych ułamkowych separatorem dziesiętnym jest '.' (kropka).

## Wejście

W pierwszej sekcji znajduje się  $v$  linii z dwoma liczbami rzeczywistymi oznaczającymi współrzędne  $x$  oraz  $y$  kolejnych wierzchołków. Po pierwszej sekcji następuje druga sekcja, w której znajduje się  $e$  linii z dwoma całkowitymi liczbami nieujemnymi oraz jedną nieujemną liczbą rzeczywistą oznaczającymi: nr wierzchołka początkowego, końcowego oraz wagę krawędzi. Liczby  $v, e$  należą do zbioru liczb naturalnych – program musi umożliwiać wczytanie dowolnie dużej ilości krawędzi i wierzchołków (kontener przechowujący graf musi zapewniać dynamiczną alokację).

## Wyjście standardowe

Wyjście ma zawierać  $n$  linii (dla każdego cyklu jedna linia). Każda linia składa się z całkowitych liczb nieujemnych (numery kolejnych wierzchołków) i jednej nieujemnej liczby rzeczywistej (waga). Wszystkie wartości oddzielone są pojedynczym znakiem spacji. Wierzchołki muszą być wypisane w takiej kolejności, w jakiej znajdują się w cyklu.

## UWAGA!

Proszę nie robić żadnych interfejsów tekstowych np. pytających się o kolejne wierzchołki, po prostu czytamy ze strumienia dane i do strumienia piszemy, tak żeby możliwe było przekierowanie plików do i z programu.

## Przykład działania:

### Dane wejściowe (cin):

```
2.0 4.0
3.0 4.0
2.0 3.0
3.0 3.0
1.5 1.0
3.0 1.0
0 1 10.0
0 2 3.3
0 3 20.4
1 3 5.0
```

2 1 3.0  
2 5 15.1  
3 4 8.0  
4 5 7.0  
5 4 3.0  
5 0 4.5

**Dane wyjściowe (cout):**

0 2 1 3 4 5 0 30.8  
0 3 4 5 0 39.9  
4 5 4 10.0