

Egy nemzeti parkban minden nevezetességhez el lehet jutni kerékpáron is. Egy kemény tél után azonban egyes kerékpárutak megrongálódtak. Minden utat felmérték, és meghatározták annak *felújítási értékét*, vagyis hogy mennyire lenne fontos azt felújítani.

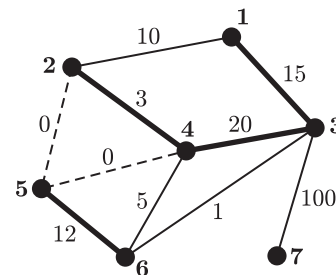
A park vezetősége úgy döntött, hogy még a nyári csúcsszezon előtt a legfontosabb felújításokat el kell végezni, az viszont nem fordulhat elő, hogy a látogatók a felújítások alatt ne tudjanak minden nevezetességhez kerékpárral is eljutni. Egyszerre tetszőlegesen sok útszakasz újítható fel, ugyanakkor minden útszakasz felújítása az egész tavaszi időszakot igénybe veszi.

Írjunk programot, mely meghatározza, hogy a tavasz során legfőljebb hány felújítás végezhető el, és ez alapján mennyi a ténylegesen felújított utak felújítási értékei összegének maximuma.

A program az utak leírását fájlból olvassa, az eredményt fájlba írja. A bemeneti, illetve kimeneti fájlok nevei az első, illetve második parancssori argumentum (az alábbi példában `s32.be.txt` `ki.txt`).

A bemenet első sorában egy-egy szóközzel elválasztva a nevezetességek N (legfőljebb 100) és az őket összekötő kerékpárutak M (legfőljebb 500) száma található. A nevezetességeket az $1, 2, \dots, N$ számokkal azonosítjuk. Az ezt követő M sor mindegyike három pozitív egész számot tartalmaz szóközzel elválasztva $A \ B \ F$ alakban: egy-egy kerékpárút által (kétirányúan) összekötött nevezetességek A és B sorszámát, valamint annak $F > 0$ felújítási értékét (1 és 1000 közötti egész), illetve $F = 0$, ha az utat nem kell felújítani.

Példa be.txt	Példa ki.txt
7 10 1 2 10 1 3 15 2 4 3 2 5 0 3 4 20 3 6 1 3 7 100 4 5 0 4 6 5 5 6 12	50



A kimenet egyetlen szám: az optimális felújítás esetén felújított utak felújítási értékeinek összege.

Beküldendő a program forráskódja (`s32.pas`, `s32.cpp`, ...), valamint a program rövid dokumentációja (`s32.txt`, `s32.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztő környezetben fordítható.