

Egy vállalat N embert foglalkoztat. Mindenkinek legfeljebb két beosztottja (akiknek a sorrendje nem fontos), és pontosan egy felettese van, kivéve a főigazgatót, akinek nincs felettese. A munkatársakat az $1, \dots, N$ számokkal azonosítjuk, az 1-es a főigazgató. A cég súlyos pénzügyi gondokkal küzd, ezért elhatározták, hogy átszervezik a beosztás rendszerét. Ehhez felvehetnek új munkatársakat, illetve megválhatnak bizonyos alkalmazottaktól. Ezenkívül az egyedüli megengedett művelet az azonosítószámok átírása. Minden egyes új munkatárs felvételéhez állásinterjút kell szervezni F krajcárért, valamint tudjuk, hogy az i -edik alkalmazott elbocsátása a végkielégítés miatt $V[i]$ krajcárba kerül. Az azonosítók átírásának természetesen nincs költsége.

A cégvezetés megadta, hogy szeretnénk mi az optimális vállalati struktúra. A fenti feltételek erre a szerkezetre is igazak. Írjunk programot, ami megadja, hogy legalább hány krajcárt kell elkölteni az átalakításra.

A standard bemenet első sora az alkalmazottak N számát és egy új alkalmazott felvételének F költségét tartalmazza. A második sor N egész számot tartalmaz, az i -edik szám $V[i]$, vagyis az i -edik alkalmazott végkielégítése. A harmadik sor $N - 1$ számot tartalmaz, az i -edik szám az $(i + 1)$ -edik beosztott felettesének a száma. A negyedik és ötödik sor az átalakítás után elérni kívánt struktúrát írja le: a negyedikben az alkalmazottak kívánt M száma található, míg az ötödikben $M - 1$ egész szám írja le a feletteseket a harmadik sorhoz hasonló módon.

A standard kimenet első és egyetlen sora tartalmazza az átalakítás minimális K költségét.

Korlátok: $1 \leq N, M \leq 5000$; $0 \leq V[i], F \leq 10^5$; időlimit: 0,2 mp; memórialimit: 256 MB.

A verem (stack) méretére nincs külön korlát, ez is a teljes memóriába számít bele.

Bemenet (a / jel a mintában újsor karaktert jelent)	Kimenet
5 1 / 6 5 4 3 2 /	4
1 1 3 3 / 6 / 1 1 2 4 4	

Magyarázat: elbocsátjuk a 5-ös számú alkalmazottat, és a 4-es számú alá felveszünk két új beosztottat. Ekkor a költség $V[5] + 2 \cdot F = 2 + 2 \cdot 1 = 4$. Ha átnevezzük a 2-est 3-assá, a 3-ast 2-essé, akkor megkapjuk a kívánt struktúrát.

Pontozás és korlátok: a programhoz mellékelt, a helyes megoldás elvét tömören, de érthetően leíró dokumentáció 1 pontot ér. A programra akkor kapható meg a további 9 pont, ha bármilyen hibátlan bemenetet képes megoldani a fenti korlátoknak megfelelően.

Beküldendő egy tömörített `s109.zip` állományban a program forráskódja, valamint a program rövid dokumentációja, amely a fentiekén túl megadja, hogy a forrás mely fejlesztői környezetben fordítható.