

Szeretnénk a számítógépünket felújítani, ezért vásárolnunk kell bele néhány alkatrészt, így a régieket az újakra tudjuk cserélni. A lehető legolcsóbban szeretnénk a felújítást elvégezni, így a legkevesebb alkatrészt szeretnénk megvásárolni (mindegyik azonos árban van). Viszont előfordulhat, hogy valamilyen összetartozó k db alkatrész közül $k - 1$ -et újra cserélünk, ekkor kötelezően a k -adikat is ki kell cserélni. A boltban összesen N ($1 \leq N \leq 1\,000\,000$) típusú alkatrész található (1-től N -ig számozva). Az összetartozó csoportok mérete bármekkora lehet, de összes méretük legfeljebb 250 000 lehet. Tudjuk továbbá azt is, hogy az 1-es számú alkatrészt mindenképp ki kell cserélnünk.

A program olvassa be a standard input első sorából N -et és C -t (a csoportok számát), majd a következő C sorból a csoportban lévő alkatrészek db_i számát, majd a csoportban lévő alkatrészek típusát db_i darab 1 és N közötti egész formájában. A két csoportnak akár közös elemei is lehetnek. A program írja a standard output első sorába a megvásárolandó alkatrészek minimális számát. Magyarázat: a példában az 1, 2, 3, 4-es számú alkatrészeket kell megvenni.

Példa bemenet:	Példa kimenet:
10 4	4
2 1 3	
2 3 4	
6 1 2 3 4 6 7	
4 4 3 2 1	

Pontozás és korlátok: A programhoz mellékelte 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 az 1 mp futásidőkorláton belül.

Beküldendő egy tömörített `s98.zip` állományban a program forráskódja (`s98.pas`, `s98.cpp`, ...) az `.exe` és más, a fordító által generált állományok nélkül, valamint a program rövid dokumentációja (`s98.txt`, `s98.pdf`, ...), amely a fentiekén túl megadja, hogy a forrás mely fejlesztői környezetben fordítható.