Atommeghajtású űrhajónkkal szeretnénk eljutni egy másik csillagrendszerbe úgy, hogy a lehető legtöbb titánnal érkezzünk meg. Van egy térképünk az ismert csillagrendszerekről, melyek között kizárólag egyirányú féregjáratokon tudunk közlekedni. A féregjáratrendszer jelenlegi állapotában teljesül minden csillagrendszerre, hogy ha kilépünk belőle, akkor biztosan nem tudunk visszajutni.

Minden csillagrendszerről tudjuk, hogy mennyi uránt (melyet üzemanyagként használunk), illetve titánt tudunk bányászni, ha meglátogatjuk. Az urán tárolása körülményes, ezért csak meghatározott mennyiséget tudunk belőle raktározni. Minden féregjáratról tudjuk, hogy mennyi uránt fogyasztunk az áthaladás közben. Szerencsére bármely csillagrendszerben feltölthetjük uránkészletünket egy egységnyi titánért. Kezdetben tele van az üzemanyagtartályunk, és nincs titánunk.

Írjunk programot, amely a standard inputról beolvassa a térképet, valamint az induló- és a célcsillagrendszer sorszámát, és megadja, hogy legföljebb mennyi titánnal érhetünk el a célba, továbbá az ehhez szükséges útvonalat is.

Bemenet: A standard input első sorában öt szám található, szóközökkel elválasztva: a csillagrendszerek ($2 \leq N \leq 10\,000$) száma, a féregjáratok ($1 \leq M \leq 200\,000$) száma, az induló- és a célcsillagrendszer sorszáma (a csillagrendszereket 1-től N-ig sorszámozzuk), valamint az üzemanyagtartályunk ($1 \leq K \leq 1\,000\,000$) kapacitása.

A következő N sor rendre az $1, 2, \ldots, N$-edik csillagrendszert írja le két, szóközzel elválasztott egész számmal, melyek megadják a bányászható titán és urán mennyiségét ($0 \leq T, U \leq 1\,000\,000$). Az utolsó M sor rendre az $1, 2, \ldots, M$-edik féregjáratot írja le három egész számmal, melyek a járat kezdő- és célcsillagrendszerének sorszáma és az áthaladáshoz szükséges urán mennyisége ($0 \leq W \leq 1\,000\,000$).

Feltehetjük, hogy a kezdő- és célcsillagrendszer nem esik egybe, valamint, hogy nincs két olyan féregjárat, melyek ugyanazt a két csillagrendszert kötik össze, továbbá, hogy a féregjáratok kezdő- és végpontja sem esik egybe.

Kimenet: A program írja ki a standard kimenet első sorába, hogy maximum mennyi titánnal érkezhetünk meg célunkhoz. A második sorba írjunk ki egy ehhez szükséges útvonalat is, a következő formában: az első szám legyen az útvonalban szereplő csillagrendszerek száma (beleértve a kezdő- és célcsillagrendszert), aztán az úton szereplő csillagrendszerek sorszámának felsorolása következzen.

Amennyiben nem juthatunk el a célig, akkor a standard kimenet egyetlen sorába a `-1` szám kerüljön.

Példák:

Bemenet	**Kimenet**	**Bemenet**	**Kimenet**
2 1 1 2 5 1 1 2 3 1 2 4	3 2 1 2	3 3 1 3 5 2 0 2 0 2 0 1 2 0 2 3 0 1 3 0	6 3 1 2 3
4 4 1 4 5 0 0 2 5 0 0 3 5 1 2 6 1 3 3 3 4 3 2 4 1	-1	5 6 1 4 10 0 0 1 0 1 0 0 0 0 3 1 2 9 1 5 3 5 3 6 3 2 1 2 4 2 3 4 2	2 5 1 5 3 2 4

Pontozás: A programhoz mellékelt, a helyes megoldás elvét tömören, de érthetően leíró dokumentáció 2 pontot ér. A programra akkor kapható meg a maximális 8 pont, ha bármilyen, a feltételeknek megfelelő tesztesetet képes megoldani a 3 mp futási időlimiten belül. Kapható részpontszám, ha a program csak kisebb tesztesetekre tud lefutni időben, *továbbá akkor is, ha a program csak olyan teszteseteket tud megoldani, amelyeknél a féregjáratokon való áthaladáshoz szükséges üzemanyag mennyisége mindig* 0.

Beküldendő egy tömörített `s74.zip` állományban a program forráskódja (`s74.pas`, `s74.cpp`, ...) az `.exe` és más, a fordító által generált állományok nélkül, valamint a program rövid dokumentációja (`s74.txt`, `s74.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrás mely fejlesztői környezetben fordítható.