

Egy házat szeretnénk felépíteni, ahol ismertek a munkafázisok (pl. üvegezés, villany bevezetése, vakolás stb.). Az építési munkákról ismert, hogy egyes feladatok elkezdése után legalább mennyi időnek kell eltelnie, amikortól bizonyos más feladatokat elkezdhetünk (pl. a villany bevezetése előtt meg kell építeni a falakat, ami  $x$  nap). Egyszerre akár több munkafázison is dolgozhatunk. A ház építéséhez bérelnünk kell eszközöket az utolsónak elkezdett munkafázis első napjáig, mivel feltehetjük, hogy a legutolsó munkafázis egy nap alatt befejezhető. Tehát ha a  $T$ . napon kezdünk el utoljára munkafázist, akkor összesen  $T \cdot P$  forint bérleti díjat kell fizetni. Az alapanyagok megvásárlása következtében minden munkafázis további költségeket jelent. A mostani feladatban úgy tekintjük, hogy az alapanyagok ára monoton csökken (azaz későbbi időpontban sosem lesz nagyobb, mint egy korábbi esetén lenne), és intervallumonként konstans. (Vagyis pl. 1–10. napig 30 forint, 11–20. napig 20, utána 21–40. napig 15 forint.)

Írjunk programot, amely a feladatok egymásra épülése, a  $P$  bérleti díj, illetve az egyes feladatokhoz szükséges intervallumonkénti alapanyagárak ismeretében megadja, hogy minimum mennyi pénzbe kerül a ház fölépítése, ha a lehető legügyesebben ütemezzük az egyes feladatok elkezdését.

A standard bemenet első sorában található a feladatok  $N$ , illetve az egymásra épülések  $E$  száma, valamint a  $P$  bérleti díj ( $1 \leq N \leq 30\,000$ ,  $0 \leq E \leq 100\,000$ ,  $0 \leq P \leq 1000$ ) szöközőkkel elválasztva. A következő  $E$  sor mindegyikében három szám található, példaként  $(a_i, b_i, c_i)$ , amelyek egymásra épüléseket írnak le: az  $a_i$  sorszámú munka megkezdése után  $c_i$  nappal lehet elkezdeni a  $b_i$  sorszámú munkát ( $1 \leq i \leq E$ ). A következő  $N$  sor az egy-egy munkához szükséges alapanyagok árainak alakulását tartalmazza a következőképpen: az első szám  $K$  ( $1 \leq K \leq 100$ ), és ezután  $K$  darab egymáshoz illeszkedő intervallum leírása következik két-két számmal  $(f_j, v_j)$ , melyek jelentése: az intervallumon  $f_j$  forintba kerül megvenni az alapanyagokat az adott munkához, és a  $v_j$ -edik nap az intervallum utolsó napja (azaz amikor még  $f_j$  forintba kerül a vásárlás) ( $0 \leq f_j \leq 1000$ ,  $1 \leq v_j \leq 10^9$ ). Az első intervallum az 1. napon indul, a továbbiak pedig mindig az előző intervallum vége utáni napon kezdődnek ( $1 \leq j \leq K$ ).

A standard kimenet egyetlen sorába írjuk ki a ház fölépítésének minimális költségét.

Feltehetjük, hogy az egymásra épülésekből nem következik ellentmondás, vagyis a ház mindenképpen felépíthető. Az egyes feladatokhoz szükséges alapanyagokat a feladat elkezdésének napján kell megvennünk. Az utolsó árintervallumok minden feladatnál ugyanazon a  $D$ . napon végződnek ( $1 \leq D \leq 10^9$ ). Az építkezést leghamarabb az 1. napon kezdhetjük, és be kell fejeznünk legkésőbb a  $D$ . napon.

Példa bemenet:	Példa kimenet:
4 5 1	19
1 2 1	
2 4 1	
3 4 4	
1 3 2	
1 4 2	
2 3 4 1 10	
1 2 10	
1 3 10	
2 10 7 3 10	

Magyarázat a példához: Egy optimális építési tervben pl. a következők lehetnének az elkezdési idők: 1, 2, 4, 8 vagy 1, 2, 3, 8. Ekkor az alapanyag beszerzések költsége  $3 + 2 + 3 + 3 = 11$ , és ehhez jön még 8 Ft bérleti díj, vagyis az összköltség 19. (Ha az elkezdési idők pl. 1, 2, 3, 6 lennének, akkor az összköltség  $3 + 2 + 3 + 10 + 6 = 24$  lenne.)

**Értékelés:** A programhoz mellékelte, a helyes megoldás elvét tömören, de érthetően leíró dokumentáció 1 pontot ér. A programra akkor kapható meg a maximális 9 pont, ha bármilyen, a feltételeknek megfelelő tesztesetet képes megoldani 1 mp futásidőkorláton belül. A megoldásra részpontszám kapható, ha a program csak a maximális bemeneti számoknál nagyságrendekkel kisebb tesztesetekre tud lefutni időben (pl. kisebb  $N$ ,  $E$ ,  $v_j$ , vagy  $K$ ).

Beküldendő egy tömörített `s81.zip` állományban a program forráskódja (`s81.pas`, `s81.cpp`, ...) az `.exe` és más, a fordító által generált állományok nélkül, a program rövid dokumentációja (`s81.txt`, `s81.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrás mely fejlesztői környezetben fordítható.