

Leveringstid / Delivery time

Mirkos Pizzeria er det bedste i byen. Det er så godt, at alle beboerne spiser pizza til frokost hver dag. Mirkos leveringsservice er så hurtigt, at man ikke opdager leveringstiden. Problemet er at bage pizzaerne hurtigt nok, da alle beboerne har deres egen favoritpizza, så det tager forskellige mængder tid for hver pizza. Mirko har kun én lille stenovn med plads til én pizza af gangen, så god planlægning er ekstremt vigtigt og må gøres inden dagen starter. For hver af de N beboere (betegnet med tallene 1 til N) kender vi bagetiden for deres yndlingspizza (T_i), og det tidspunkt de spiser frokost (L_i). Hvis en beboer modtager deres pizza K tid inden deres frokost, så modtager Mirko en bonus på K kroner. Modsat må Mirko betale K kroner for den mængde tid (K) pizzaen er forsinket (pga. hans pizzaleveringsgaranti). Hvis pizzaen ankommer præcis til tiden modtager og mister Mirko intet.

Mirko vil gerne kende den maksimale bonus (inklusive alle tab pga. garantien), som han kan tjene på en dag, hvis pizzaerne bliver bagt i den mest effektive rækkefølge. Læg mærke til at Mirko kan tjene en negativ bonus, hvis han skal betale mere end han modtager.

Siden beboerne nogle gange ændrer deres favoritpizza og deres frokosttid, må Mirkos pizzabagningsplan løbende opdateres for at tjene den optimale bonus. Skriv et program, som udregner den maksimale totale bonus for de første krav samt efter hver ændring.

Note: I denne by starter dagen når $t = 0$, og den varer meget længere end det tager at bage pizzaer til alle beboerne. Planen for pizzabagningen (inkl. ændringer) skal lægges inden dagen starter.

Input

Den første linje input indeholder to positive integers: N (antallet af beboere) og C (antallet af ændringer til pizzafyld).

Hver af de N linjer indeholder to positive integers: L_i (beboerens frokosttid) og T_i (tiden det tager at bage deres pizza).

Hver af de næste C linjer indeholder tre positive integers: R (beboerens index), L (beboerens nye frokosttid), og T (den tid, det tager at bage deres nye favoritpizza).

Constraints:

$1 \leq N, C \leq 200\,000$,
 $0 \leq L_i, L \leq 100\,000$,
 $1 \leq T_i, T \leq 100\,000$,
 $1 \leq R \leq N$.

Output

Den første linje output skal indeholde den maksimale totale bonus for de første krav fra beboerne. For hver af de C ændringer skal outputtet indeholde endnu en linje, som indeholder den nye maksimale bonus efter ændringen.

Sample input	Sample output
3 2 10 2 6 5 4 3 1 6 1 3 0 10	3 2 -11
4 2 3 2 0 3 4 3 4 1 3 0 4 1 4 5	-8 -13 -18
6 7 17 5 26 4 5 5 12 4 8 1 18 2 3 31 3 4 11 5 4 19 3 5 23 2 6 15 1 5 19 1 3 10 4	27 59 56 69 78 81 82 58

Forklaring af den første test case

Den optimale pizzabagningsplan er (1, 3, 2). På den måde vil den første pizza være færdig til tiden $t = 2$, den tredje til $t = 5$, og den anden til $t = 10$. Den første pizza vil blive leveret $K = 8$ tidligt, den anden $K = -1$ sent, og den tredje $K = -4$ sent. Den maksimale bonus bliver dermed 3 kroner.

Når den første beboer ændrer sine krav, forbliver den optimale plan uændret, mens bonusserne ændrer sig til 5, 0 og -3.

Efter den anden ændring bliver den optimale plan (1, 2, 3), mens bonusserne bliver 5, 0 og 16.