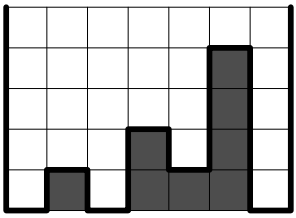
Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb a přiděleného počítače (tzn. knihy, kalkulačky, mobily, apod.).

Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Řešení odevzdáváte pomocí soutěžního systému CMS, který ho automaticky otestuje na připravených sadách testovacích dat. Za každou úlohu můžete získat až 10 bodů. Detaily hodnocení naleznete v letáku s popisem prostředí. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

P-III-4 Kaňon

Program:	`kanon.pas` / `kanon.c` / `kanon.cpp`
Vstup a výstup:	standardní vstup a výstup
Časový limit:	najdete v soutěžním systému
Paměťový limit:	najdete v soutěžním systému

V dvojrozměrném světě průzkumníci nedávno objevili novou atrakci: obrovský kaňon. Příklad takového kaňonu vidíte na obrázku:



Zobrazený kaňon má šířku $n = 7$. Jeho dno můžeme popsat posloupností výšek $(0, 1, 0, 2, 1, 4, 0)$. Každé číslo udává, kolik čtverečků země se v daném sloupci nachází. Nalevo a napravo od krajních sloupců jsou stěny kaňonu. Ty jsou svislé a dostatečně vysoké.

Nad kaňonem občas prší. Zajímavé je, že vždy prší jenom v některém jednom sloupci. Voda z oblaků padá rychlostí 1 čtvereček za sekundu. To znamená, že za každou sekundu deště spadne tolik vody, kolik zaplní jeden čtvereček na obrázku kaňonu. Voda, která dopadne na dno kaňonu, se spojitě rozlévá do stran a stéká z vyšších míst na nižší.

Soutěžní úloha

Je zadán popis kaňonu. Pro každý jeho sloupec zodpovězte následující otázku: Jestliže v tomto sloupci začne pršet, po kolika sekundách začne v tomto sloupci stoupat vodní hladina?

Uvědomte si, že odpověď na tuto otázku nezávisí na tom, jak přesně se voda do té doby rozlévala po okolí. Voda v konkrétním sloupci začne stoupat tehdy, když už zaplnila všechny níže položené části kaňonu v okolí.

Formát vstupu a výstupu

První řádek vstupu obsahuje jedno kladné celé číslo n. Na druhém řádku je n nezáporných celých čísel $a_1, \ldots, a_n$, která popisují dno kaňonu zleva doprava.

Na výstup vypište jeden řádek s posloupností n celých čísel: pro každý sloupec kaňonu zleva doprava odpověď na výše položenou otázku. Každá dvě čísla oddělte jednou mezerou. Bezprostředně za posledním číslem vypište znak konce řádku.

Omezení

Jednotlivé sady testovacích vstupních dat splňují následující omezení:

V sadách #1 a #2 platí $n \leq 100$ a $a_i \leq 100$.

V sadách #3 a #4 platí $n \leq 1\,000$ a $a_i \leq 10^6$.

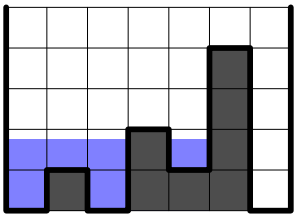
V sadách #5 až #10 platí $a_i \leq 10^9$. V těchto sadách hodnota n postupně roste od $n = 100\,000$ v sadě #5 až do $n = 2\,000\,000$ v sadě #10.

V sadách s lichým číslem jsou všechny hodnoty a_i navzájem různé.

Příklady

Vstup:	*Výstup:*
7 0 1 0 2 1 4 0	0 2 0 6 0 20 0

Příklad vstupu odpovídá kaňonu na obrázku uvedeném na začátku zadání. Na následujícím obrázku vidíte příklad toho, jak může tento kaňon vypadat poté, co 5 sekund pršelo v prostředním sloupci. Za další sekundu by se všechny okolní prohlubně zaplnily vodou úplně a potom by už začala stoupat voda také v prostředním sloupci.



Vstup:	*Výstup:*
8 0 2 0 1 0 0 1 0	0 12 0 4 0 0 4 0

Dejte pozor na to, co se děje v případě, že více úseků dna kaňonu má stejnou výšku. Když například prší ve čtvrtém sloupci, bude po 3 sekundách po jednom čtverečku vody ve třetím, pátém a šestém sloupci. Následně během čtvrté sekundy začne přetékat voda přes sedmý sloupec do osmého. Po čtyřech sekundách už bude souvislá vodní hladina sahat od třetího až po osmý sloupec. Pokud by pršelo dále, bude voda stoupat najednou v celém tomto úseku, tedy už i ve čtvrtém sloupci.

P-III-5 Demonstrace

Program:	`demonstrace.pas` / `demonstrace.c` / `demonstrace.cpp`
Vstup a výstup:	standardní vstup a výstup
Časový limit:	najdete v soutěžním systému
Paměťový limit:	najdete v soutěžním systému

V hlavním městě Absurdistánu se chystá demonstrace proti přílišnému množství demonstrací. Během demonstrace se chce dav protestujících občanů přemístit z náměstí před sídlo velkého vezíra. To ale není tak jednoduché, jak by se mohlo zdát. Hlavní problém spočívá v tom, že dav lidí nemůžete jen tak přesouvat ulicemi. Když se pokusíte poslat velký dav lidí ze široké ulice do úzké, hrozí vážné nebezpečí, že se ve vzniklé tlačenici mnozí lidé zraní.

Soutěžní úloha

Je dán popis města: počet n křižovatek, počet m ulic mezi nimi a šířka každé ulice. Každá ulice je jednosměrná a spojuje některé dvě křižovatky (v Absurdistánu nemůžete jít proti směru jednosměrky, ani když jste chodec).

Dále jsou zadány křižovatky a, b, odkud a kam je třeba dav přemístit. Jako poslední je dáno nezáporné číslo k s následujícím významem: jde-li dav ulicí šířky s, na nejbližší křižovatce může přejít jedině do ulice, jejíž šířka je alespoň $s - k$.

Napište program, který zjistí, zda je vůbec možné přemístit dav z křižovatky a na křižovatku b. Pokud ano, zjistěte, kolika nejméně ulicemi při tom dav musí projít.

Formát vstupu a výstupu

Na prvním řádku vstupu je pět nezáporných celých čísel: n, m, a, b, k. Jejich význam je vysvětlen výše. Křižovatky jsou očíslovány od 0 do $n - 1$. Následuje m řádků, z nichž každý popisuje jednu ulici. Popis každé ulice je tvořen třemi celými čísly x_i, y_i, s_i: číslo křižovatky, kde daná ulice začíná, číslo křižovatky, kde tato ulice končí, a šířka této ulice.

Na výstup vypište jeden řádek s jedním celým číslem: nejmenší d, pro které existuje řešení, v němž dav postupně projde d ulicemi. Jestliže neexistuje žádný způsob, jak dostat dav z křižovatky a na křižovatku b, vypište číslo -1.

Omezení

Ve všech sadách testovacích vstupních dat platí: $2 \leq n \wedge 0 \leq m$ (jsou alespoň 2 křižovatky a 0 ulic), $0 \leq a, b < n$, $a \neq b$ (čísla startu a cíle jsou korektní a vzájemně různá), $0 \leq x_i, y_i < n$, $x_i \neq y_i$ (čísla křižovatek jsou korektní a žádná ulice nemá oba své konce na stejné křižovatce), $0 < s_i$ (ulice mají kladné šířky).

V prvních třech sadách platí $n, m, k, s_i \leq 100$.

V sadách #4 až #6 platí $n \leq 100\,000$, $m \leq 500\,000$, $k, s_i \leq 20$.

V sadách #7 až #10 platí $n \leq 100\,000$, $m \leq 500\,000$, $k, s_i \leq 10^9$.

V sadách #7 a #8 navíc platí, že se na každé křižovatce setkává nejvýše 20 ulic.

Příklady

Vstup:	*Výstup:*
3 2 0 2 7 0 1 20 1 2 10	-1

Máme 3 křižovatky a 2 ulice. Chceme jít z křižovatky 0 na křižovatku 2. Konstanta k je 7. První ulice vede z křižovatky 0 na křižovatku 1 a má šířku 20. Druhá ulice vede z 1 na 2 a má šířku 10. Rozdíl mezi šířkami ulic je ale příliš velký, dav přicházející první ulicí nemůže pokračovat druhou ulicí, tudíž řešení neexistuje.

Vstup:	*Výstup:*
4 4 0 2 7 0 1 20 1 2 10 1 3 14 3 1 9	4

Dav postupně projde první, třetí, čtvrtou a druhou ulicí. Všimněte si, že křižovatkou 1 projde dvakrát.

P-III-6 Obdélníkový tetris

Program:	`tetris.pas` / `tetris.c` / `tetris.cpp`
Vstup a výstup:	standardní vstup a výstup
Časový limit:	najdete v soutěžním systému
Paměťový limit:	najdete v soutěžním systému

Roman by si chtěl naprogramovat hru Tetris. Pro začátek se rozhodl pro značně zjednodušenou verzi. V jeho hře se nebude z hracího plánu nikdy nic odstraňovat. Když jednou nějaký dílek někam dopadne, zůstane tam celý až do konce hry. Padající dílky se také nebudou otáčet ani posunovat. Jakmile se nový dílek někde objeví, bude ve stejných sloupcích padat dolů, dokud na něco nenarazí. Navíc dílky různých nepravidelných tvarů jsou pro Romana příliš složité. V jeho hře budou padat pouze samé obdélníky.

Soutěžní úloha

Hrací plán (tzv. šachta) je obdélník, který má šířku s a výšku 10^9. Na začátku hry je celý hrací plán prázdný. Řádky a sloupce šachty číslujeme od 0, sloupce zleva doprava a řádky zdola nahoru. Během hry se postupně, jeden po druhém, objeví na vrchu šachty n obdélníků. V pořadí i-tý z nich bude mít šířku w_i políček, výšku h_i políček a nejlevějším sloupcem, v němž tento obdélník leží, bude sloupec ℓ_i.

Napište co nejefektivnější program, který bude simulovat padání těchto obdélníků.

Formát vstupu a výstupu

Na prvním řádku vstupu jsou uvedena dvě celá čísla: šířka šachty s a počet padajících obdélníků n. Na každém z následujících n řádků jsou tři celá čísla představující parametry jednoho z obdélníků: w_i, h_i a ℓ_i.

Pro každý obdélník vypište na výstup jeden řádek s jedním celým číslem. Tím bude číslo řádku hracího plánu, kde po dopadu skončí spodek příslušného obdélníku.

Omezení

Jednotlivé sady testovacích vstupních dat splňují následující omezení:

- V sadách #1 až #3 platí $s \leq 100$, $n \leq 100$ a součet všech h_i nepřekročí 100.
- V sadách #4 a #5 je $s \leq 1\,000$ a $n \leq 100\,000$.
- V dalších sadách se hodnoty s a n postupně zvyšují, přičemž v poslední sadě platí $s \leq 2\,000\,000$ a $n \leq 1\,000\,000$.

Ve všech testovacích vstupech navíc platí:

- $1 \leq w_i$ (obdélníky mají kladné rozměry),
- $\ell_i + w_i \leq s$ (žádný nepřečnívá bokem ze šachty),
- $1 \leq h_i \leq 1\,000$ (všechny dohromady se určitě vejdou do šachty).

Příklad

Vstup:	*Výstup:*
10 7	0
1 1 4	1
4 2 3	0
2 2 8	3
2 2 6	2
2 2 8	0
1 5 2	5
10 1 0	

Obsah šachty po skončení této ukázkové hry: