



## Úlohy 2. kola letnej časti

Termín odoslania riešení tohto kola je pondelok 22. mája 2017.

### 1. Zábava musí byť

kat. Z; 6 b za popis, 4 b za program

Pilný študent Slavo sa rozhodol dať si pauzu od každodenného kolobehu a vybral sa do zábavného parku. Zistil totiž, že tam budú mať aj jeho obľúbenú disciplínu – strieľanie zo zbraní. Slavo je už starý harcovník, čo sa (aj) strieľania týka a je teda odhodlaný vyhrať akúkoľvek súťaž, čo sa tam bude organizovať.

Keď Slavo prišiel do parku, hneď sa zastavil pri tej najväčšej tabuli široko ďaleko:

**Chceš byť ako kovboj z Divokého Západu? Vystrieľaj si výhru!**

No, musíte uznať, že to<sup>1</sup> znie dosť dobre.

Vybral sa teda na strelnicu. Zistil, že sa strieľa z pušky a z brokovnice. S oboma už mal tú česť, takže v tom by problém nemal byť. Ba čo viac, všimol si, že vzdialenosť medzi jednotlivými terčmi je taká, že na jeden výstrel brokovnice dokáže trafiť aj dva terče vedľa seba. Už to len nedokašľať!

Terče na strelnici sú usporiadané zvláštnym spôsobom. Z pozície, odkiaľ sa strieľa, vidí strelec  $n$  terčov umiestnených v pravidelných rozstupoch vedľa seba. Za každým z týchto terčov môže (ale nemusí) byť ešte niekoľko iných terčov, akoby v zástupe. Ich počet je strelcovi dopredu známy. Z vtáčej perspektívy to celé môže vyzeráť napríklad takto:



Keď sa Slavo spýtal na pravidlá, dozvedel sa nasledovné:

- Terče sú labilné, teda pri zasiahnutí terču tento (tieto) padne<sup>2</sup> a odkryje sa ďalší za ním (ak tam ešte nejaký je).
- Strelec má k dispozícii neobmedzené množstvo nábojov do oboch zbraní a môže ich ľubovoľne kombinovať a striedať.
- Cieľom hry je zhodiť všetky terče na čo najmenej výstrelov.

Ako má Slavo skombinovať pušku s brokovnicou aby potreboval čo najmenej výstrelov? Kam strieľať čím? Chce vyhrať. Musí vyhrať. Chce byť ako z Divokého Západu. Pomôžte mu!

### Úloha

Pre dané počty terčov v jednotlivých zástupoch zistíte, koľko výstrelov bude Slavo potrebovať, aby vyhral. Zaujímá nás teda najmenší možný počet výstrelov, na ktorý vie Slavo zasiahnuť (a zhodiť) všetky terče.

Predpokladajte, že **Slavo nikdy neminie cieľ**. Puškou vie trafiť ľubovoľný terč, na ktorý vidí (nezakrýva ho iný terč). Brokovnicou dokáže trafiť ľubovoľné dva terče stojace v susedných zástupoch (oba terče musí byť vidno).

<sup>1</sup>Až na toho kovboj-a.

<sup>2</sup>Toto platí duplom, keďže strieľa Slavo.

## Formát vstupu

V prvom riadku je jedno celé číslo  $n$  ( $1 \leq n \leq 100\,000$ ) – počet radov terčov (teda počet terčov viditeľných na začiatku). Na ďalšom riadku je  $n$  medzerou oddelených kladných celých čísel  $1 \leq t_i \leq 10^9$  –  $i$ -te číslo udáva počet terčov v  $i$ -tom rade (dokopy, vrátane toho terča, ktorý vidíme).

## Formát výstupu

Na výstup vypíšte jediné celé číslo – počet výstrelov, ktoré Slavo vykoná pri najlepšej možnej stratégii.

Dávajte si pozor na veľkosť čísel, s ktorými pracujete. Všimnite si, že keď budete napríklad sčítavať čísla zo vstupu, výsledok sa vám nemusí zmestiť do 32-bitovej premennej. Odporúčame preto použiť `long long` v C++ a `Int64` v Pascale.

## Príklady

vstup	výstup
<pre>3 1 6 1</pre>	<pre>6</pre>

*Vstup predstavuje 3 rady terčov. V prvom a treťom rade je po jednom, v druhom rade je 6 terčov za sebou. Správna stratégia je napríklad vystreliť najprv puškou 4-krát do druhého radu a potom brokovnicou do prvých dvoch a druhých dvoch. Existujú aj iné, rovnako dobré stratégie.*

vstup	výstup
<pre>5 5 1 8 5 5</pre>	<pre>18</pre>

*Tento vstup zodpovedá obrázku v zadaní. Slavo strelí 8-krát puškou do stredného radu. Následne si vymení zbraň a 5-krát to brokovnicou napáli naraz na posledné dva rady. Zase si vymení zbraň, a 4-krát vystrelí na prvý rad. Nakoniec poslednýkrát vystrelí brokovnicou na prvý. Lepšie to nejde, no možností je opäť viac.*

vstup	výstup
<pre>6 1 5 5 5 5 3</pre>	<pre>13</pre>

## 2. Zákerní súrodenci

kat. Z; 6 b za popis, 4 b za program

Alžbetka má malú sestru. Jedného dňa sa Alžbetka rozhodla vyrobiť náhrdelník z korálok. Jej malá sestra pri tom samozrejme nemohla chýbať a musela jej pomôcť. Na začiatku Alžbetka navliekla na šnúрку nejaké korálky. Podľa jej sestry to však spravila úplne zle a teda musí vymeniť túto korálku za novú, a tamtú tiež...<sup>3</sup> Veď poznáte malé sestry.

Alžbetka by bola rada, keby ten náhrdelník vyzeral aspoň trochu dobre. Napríklad aby bol symetrický. Pri všetkom tom vymieňaní korálok by teda chcela vedieť, kedy sa tak stane.

### Úloha

Jednotlivé druhy korálok budeme reprezentovať malými písmenami anglickej abecedy **a-z**. Na začiatku dostanete popis náhrdelníka – nejaký reťazec znakov, ktorý popisuje jednotlivé korálky v náhrdelníku. Postupne v ňom budeme vymieňať korálky za iné. Po každej takejto výmene povedzte, či je daný reťazec symetrický – tj. či sa odzadu číta rovnako ako spredu.

### Formát vstupu

V prvom riadku vstupu je reťazec dĺžky  $n$  ( $1 \leq n \leq 100\,000$ ) – popis pôvodného náhrdelníka. V druhom riadku je číslo  $q$  ( $1 \leq q \leq 100\,000$ ) – počet zmien, ktoré nastanú. Ďalších  $q$  riadkov bude obsahovať medzerou oddelené číslo a znak – pozícia korálky, ktorú treba vymeniť (číslujeme od 0) a druh (písmeno), na ktorý sa má korálka zmeniť.

### Formát výstupu

Na výstup vypíšte  $q$  riadkov, pričom každý bude obsahovať buď „**ano**“ alebo „**nie**“ (bez úvodzoviek) podľa toho, či je reťazec po danej zmene symetrický, alebo nie.

<sup>3</sup>Áno, správne ste uhádli. Alžbetka musí vždy vyvliecť väčšinu korálok aby sa k tej “zlej” korálke vôbec dostala. A potom ich dať zase späť. Ale to v tejto úlohe vôbec nie je podstatné.

Pozor na veľkosť vstupu a výstupu. S pomalým načítavaním/vypisovaním môžete dostať hlášku „TLE - Prekročený časový limit“ aj s inak rýchlym riešením.

### Príklady

vstup	výstup
abcabc 3 0 c 2 a 3 c	nie ano nie

Po prvej zmene bude reťazec vyzeráť takto: cbcabc, čo nie je symetrické. Po druhej zmene bude vyzeráť takto: cbaabc, čo symetrické je. Treťou zmenou nám to sestra pokazí a reťazec opäť nebude symetrický.

vstup	výstup
PPPPP 2 2 a 3 p	ano ano

Prvou zmenou sme zmenili znak v strede; reťazec ostane symetrický. Druhou zmenou zmeníme p na p; reťazec ostane rovnaký.

## 3. Zblúdila krava

kat. Z; 6 b za popis, 4 b za program

Každý vie, aký je Denis farmár. Na svojom pozemku chová veľa rôznych zvieratiek, ako napríklad psíka Toma, zopár kôz a stádo kráv. Psík Tom má náročnú úlohu – stráži kravy, aby sa netúlali mimo Denisov pozemok (sám Denis ich nestráži, radšej pozerá futbal na svojom iPade). Tom nie je zrovna inteligentný pes. Napriek tomu je schopný postrážiť všetky zvieratká okrem kravy Rysule. Tá je veľmi múdra, a často sa jej podarí Toma prechytračiť a opustiť pozemok. Všimla si totiž, že ak sa pohybuje iba vertikálne alebo horizontálne, tak ju Tom nezbadá. Potom chudák Denis musí Rysulu hľadať a často pritom zmešká druhý polčas. Keďže sa blíži Liga majstrov, Denis si nemôže dovoliť zmeškať zápas kvôli Rysuli. Musí preto oplotiť svoj pozemok.

Denis sa rozhodol, že zabije dve muchy jednou ranou. Vyženie všetky zvieratá na pastviny, a kým sa budú pást, on bude stavať plot. Zabudol ale, že Rysula mu bude chcieť ujsť, a on už nechce dopustiť aby mu zase zdrhla.

Denis sa naučil techniku Assembler, to znamená, že jeho **runtime** (beh) je nekonečne rýchly. Plot ale stavia pomaly, konkrétne tak rýchlo, ako sa Rysula pohybuje.

### Úloha

Denisov pozemok je obdĺžnik  $r \times s$  políček. Na jednom z týchto políček sa na začiatku pasie krava Rysula, ktorá mu chce z neho ujsť. Každú sekundu sa stanú nasledovné dve veci (v takomto poradí):

- Denis postaví kus plota o dĺžke strany jedného políčka. Tento kus oddelí niektoré okrajové políčko jeho pozemku od vonkajšieho sveta (v prípade rohového políčka ho oddelí iba z jednej strany). Keďže Denis behá nekonečne rýchlo, môže túto časť plota postaviť kdekoľvek, nezávisle na tom, kde bol v predošlej sekunde.
- Rysula sa pohne o jedno políčko v jednom zo štyroch základných smerov. Ak je Rysula na kraji pozemku a nebráni jej plot, môže z neho takýmto spôsobom ujsť.

Denis vyhrá, ak sa mu podarí oplotiť celý pozemok (teda postaviť všetkých  $2(r + s)$  častí plota) bez toho, aby z neho Rysula ušla. Rysula vyhrá, ak sa jej podarí ujsť z pozemku skôr, než ho Denis celý oplotí.

### Táto úloha je interaktívna!

V tejto úlohe si váš program zahrá proti nášmu programu. Váš program bude hrať za Denisa, náš za Rysulu.

Váš program bude striedavo vypisovať ťahy, ktoré chce urobiť a načítavať ťahy, ktoré urobí ten náš, až kým niektorý program nevyhrá.

## Formát vstupu

Na začiatku hry dostanete na vstupe rozmery mapy a začiatočnú pozíciu kravy. V prvom riadku budú dve medzerou oddelené celé čísla  $r$  a  $s$  ( $1 \leq r, s \leq 20\,000$ ) – rozmery Denisovho pozemku. V ďalšom riadku sú čísla  $x$  a  $y$  – počiatočná pozícia kravy Rysule.

Následne môžete urobiť svoj prvý ťah a vypísať ho na výstup. Po každom ťahu, ktorý urobíte (a vypíšete), dostanete na vstupe náš ďalší ťah. Ten bude mať formu dvojice čísel  $x$  a  $y$  – pozícia, kam sa Rysuľa pohla. Platí  $1 \leq x \leq s$  a  $1 \leq y \leq r$ . Ak sa Rysuli podarí ujsť, váš program sa to už nedozvie (aj tak by s tým už nemal čo robiť).

Rozmery mapy a počiatočná pozícia kravy budú vždy také, že existuje víťazná stratégia pre Denisa. To znamená, že váš program by mal byť schopný vyhrať bez ohľadu na to, ako dobre bude hrať ten náš.

V jednotlivých sadách testovacích vstupov navyše platí:

číslo sady	obmedzenie na $r, s$	Rysuľa
1	$1 \leq r, s \leq 100$	Rysuľa sa nepokúsi ujsť z pozemku (na získanie bodov ho stačí korektne oplotiť).
2	$1 \leq r, s \leq 1000$	Rysuľa najviac raz príde na okraj pozemku. Ak vtedy narazí na plot, vzdá to a už sa nepokúsi utiecť.
3	$1 \leq r, s \leq 1000$	Rysuľa sa zo všetkých síl snaží ujsť.
4	$1 \leq r, s \leq 20\,000$	Rysuľa sa zo všetkých síl snaží ujsť.

Môžete predpokladať, že vstup je korektný a Rysuľa sa vždy pohne práve o 1 políčko.

## Formát výstupu

Po každom načítaní pozície Rysule vypíšte, ktorú časť pozemku chcete oplotiť. Keďže rohy pozemku sa dajú oplotiť z dvoch strán, pomôžeme si trikom. Zväčšíme Denisov pozemok o 1 políčko z každej strany. Krajné políčka budú “oplocovať” pôvodné Denisove pastviny. Takto navyše vieme každej časti plotu priradiť jednoznačné súradnice. Ak chcete napríklad oplotiť políčko  $[1, 1]$  zľava, vypíšte: `0 1`.

(*Dávajte si pozor na korektnosť vášho výstupu. Ak vypíšete súradnice plotu, ktorý neexistuje, alebo je už postavený dostanete „WA - Zlá odpoveď“.*)

## Flushovanie

Programovacie jazyky štandardne používajú výstupný buffer (väčšinou 512 bajtov), a až po jeho zaplnení program vypíše jeho obsah. Flushovanie vlastne znamená vypísanie (vyprázdnenie) tohto buffera.

Vždy, keď vypíšete svoj ťah, musíte ešte flushnúť výstupný buffer. Inak sa váš ťah v skutočnosti nevypíše a nedostane sa k nášmu programu. V konečnom dôsledku potom dostanete hlášku „TLE – Prekročený časový limit“.

## Ako flushovať?

Ak programujete v C/C++ a používate `printf()`, na to aby ste flushli jeho buffer použite `fflush(stdout)`, ak používate `cout`, buffer sa flushuje automaticky po vypísaní konca riadku pomocou `cout<<endl`. Ak programujete v Pythone, používajte `print(vystup, flush=True)`. Ak programujete v Pascale použite `flush(output)`

## Príklady

vstup	výstup
<pre>9 9 5 5 6 5 7 5 8 5 ... ďalších 32 riadkov</pre>	<pre>0 1 10 5 10 1 ... ďalších 33 riadkov</pre>

*Denis najprv postavil plot na ľavej strane rohového políčka  $[1, 1]$ . Krava sa rozbehla doprava. Denis postavil plot napravo od políčka  $[9, 5]$ , kam má Rysuľa namierené. Rysuľa sa opäť pohla doprava. Po ďalších 34 ťahoch sa Denisovi podarilo oplotiť celý pozemok.*

## 4. Zimný problém Sysľa Mariána

kat. Z a O; 9 b za popis, 6 b za program

Ako všetci viete, v našich Nízkych Tatrách v hojnom počte žijú sysle. Býva tam aj sysel Marián. Svoj príbytok si vybudoval tak, že vyhlbil na lúke priechodný, pohodlný tunel, ktorý na konci rozšíril na priedušnú, útulnú izbu. V nej opäť vyhlbil niekoľko priechodných, pohodlných tunelov, ktoré na konci rozšíril na priedušné, útulné izby. Takto pokračoval, v každej už vyhlbenej priedušnej, útulnej izbe vyhlbil smerom nadol niekoľko (nula alebo viac) priechodných, pohodlných tunelov. Keď skončil, jeho syslie obydlie tvorilo  $n$  útulných, priedušných izieb spojených  $n - 1$  priechodnými, pohodlnými tunelmi. Marián bol so svojím výtvorom náramne spokojný.

Jedného večera, zatiaľ čo riešil syslie veci v jednej zo svojich izieb, začalo vonku mrznúť. Mariánovi bolo jasné, že ak by vyliezol do niektorej izby, ktorá je bližšie k povrchu ako tá, v ktorej práve je, tak by určite prechladol. Nič si z toho však nerobil – uložil sa spať v jednej z prístupných izieb (buď ostal v tej, čo bol, alebo sa presunul niekam hlbšie). No nemohol tušiť, že kvôli ukrutnej zime primrzol neďaleký rybník. Žaba Michal sa teda rozhodol, že si nájde priedušnejšiu, útulnejšiu príbytok. A tak zamieril priamo do Mariánovej nory. Vďaka priechodnosti a pohodlnosti tunelov bez problémov doskákala až do priedušnej, útulnej izby v ktorej spal Marián. Marián sa v tú noc prebudil na neúprosne chrápanie jeho novonadobudnutého spolubývajúceho.

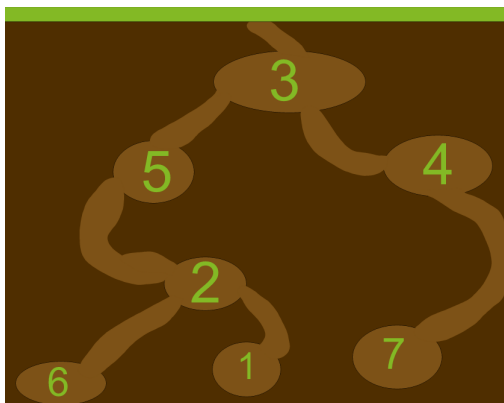
Čo má teraz robiť? Marián dobre vedel, že ak by Michala vyhnal, tak by v studenej noci určite ochorel. Na to však nemá srdce. Nechal teda žabu spať, prešiel cez priechodný, pohodlný tunel do vedľajšej priedušnej, útulnej izby a horko-ťažko predsa len zaspal. Keď sa ráno zobudil, žaby už nebolo. Marián si však domyslel, že sa táto situácia bude celú zimu opakovať – akonáhle začne večer vonku mrznúť, Marián si bude musieť nájsť takú priedušnú, útulnú izbu, do ktorej sa vie dostať pomocou priechodných, pohodlných tunelov bez toho, aby liezol smerom nahor (tam by totiž prechladol). Hneď na to k nemu doskáká žaba Michal a Mariánovi nedá svojim chrápaním spať. Teraz Marián potrebuje ujsť do inej izby čo najďalej od žaby bez toho aby vyliezol nad izbu v ktorej bol večer (tam sa mu bude najľahšie zaspávať). Marián si ďalšiu noc radšej naplánuje.

### Úloha

Izbám v svojej nore priradil Marián čísla od 1 po  $n$ . Číslo izby, ktorú vyhlbil ako prvú (teda izby najbližšie k povrchu) označme  $l$ . Hĺbka izby je počet tunelov, cez ktoré treba prejsť smerom nadol, aby ste sa do nej dostali z lúky. Hĺbka izby  $l$  je teda 1 a hĺbka každej inej izby je o 1 väčšia než hĺbka izby nad ňou.

Vzdialenosť medzi dvoma izbami je počet tunelov, cez ktoré musíme prejsť, aby sme sa dostali z jednej do druhej.

Mariánova nora môže vyzeráť napríklad takto:



Izbu, kde sa Marián zdržiava večer, označme  $v$ . Počas celej noci sa Marián nemôže ani na chvíľu ocitnúť v izbe s menšou hĺbkou, než hĺbka izby  $v$ . To znamená, že ani počas presunu z izby  $v$  do izby, kam sa prvotne uloží spať, ani počas nočného úteku od žaby nemôže Marián ani len prechádzať cez izbu s menšou hĺbkou ako má izba  $v$ . Samozrejme, aj obe izby, kde bude Marián spať, musia byť aspoň tak hlboko ako  $v$ .

Marián potrebuje o každej izbe zistiť, aká je dobrá. To znamená, že pre každú izbu ho zaujíma odpoveď na otázku “Ak by som sa večer zdržiaval v tejto izbe a izbu na spanie by som si zvolil čo najlepšie, ako ďaleko od žaby sa mi v noci podarí dostať?” Inými slovami, pre každú izbu  $x$  ho zaujíma najväčšia možná vzdialenosť medzi dvojicou izieb  $y, z$  takou, že z izby  $x$  sa vie dostať do izby  $y$  a z izby  $y$  do izby  $z$  bez toho, aby musel vyliezť do menšej hĺbky než má izba  $x$ .

Pomôžte úbohému sysľovi!

### Formát vstupu

V prvom riadku sú dve čísla  $n, l$  ( $1 \leq l \leq n \leq 150\,000$ ) – počet izieb v Mariánovej nore a číslo izby, ktorá je priamo prepojená s lúkou (a má teda hĺbku 1). Nasleduje  $n - 1$  riadkov s dvojicami čísel izieb  $a_i, b_i$ , ktoré

sú prepojené tunelom. Je zaručené, že z každej izby sa dá postupnosťou tunelov dostať do každej inej práve jedným spôsobom. Izbičky s chodbami teda tvoria strom.

V prvej sade testovacích vstupov navyše platí, že z každej izby ide najviac jeden priechodný, pohodlný tunel do izby s väčšou hĺbkou. To znamená, že nora sa nerozvetvuje, iba stále klesá dodola.

### Formát výstupu

Vypíšte  $n$  riadkov. V  $i$ -tom z nich vypíšte hľadanú vzdialenosť pre izbu  $i$ , ako je popísané v časti Úloha.

### Príklad

vstup	výstup
4 2	0
2 3	3
4 1	2
3 4	1

*Toto je príklad vstupu z prvej sady.*

vstup	výstup
7 3	0
4 7	2
2 1	5
6 2	1
5 2	2
3 4	0
5 3	0

*Toto je syslia nora z obrázku v Úlohe. Ak je Marián večer v niektorej z izieb 1, 6 alebo 7, má smolu a musí v nej ostať po celú noc (teda bude od žaby vzdialený nula). Ak je večer v izbe 2, môže sa uložiť spať do izby 6 a počas noci utiecť do izby 1 (vzdialenosť 2). Ak je večer v izbe 3, môže ísť spať napríklad do izby 1 a v noci sa presunúť do izby 7 (vzdialenosť 5). Ak začína v izbe 4, môže ostať spať v nej a v noci utiecť do izby 7 (vzdialenosť 1). Ak bude večer v izbe 5, môže sa uložiť v izbe 6 a v noci prejsť späť do izby 5 (vzdialenosť 2).*

## 5. Obrovská rekonštrukcia

kat. Z a O; 8 b za popis, 7 b za program

Ako už viete z tretej úlohy v predošlej sérii<sup>4</sup>, starosta Kocúrkova plánuje rekonštrukciu autobusových zastávok.

Na pripomenutie, Kocúrkovo je jedna dlhá ulica pozdĺž ktorej je postavených  $n$  autobusových zastávok. Každá autobusová linka má dve konečné zastávky, medzi ktorými jej autobusy jazdia hore-dolu, pričom stoja aj na každej zastávke po ceste. Keďže Kocúrkovo je Kocúrkovo, môže sa stať, že niektorá linka začína aj končí na rovnakej zastávke (a teda autobus iba stojí na tejto zastávke).

Z politických dôvodov (o ktorých sa môžete dočítať v zadaniach minulej série) starosta chce, aby od začiatku rekonštrukcie čo najdlhšie platilo, že každý deň sa na každej linke rekonštruuje aspoň jedna zastávka.

Starosta vyhlásil verejnú súťaž a dostal veľa ponúk. Teraz potrebuje zistiť, ktorá z nich je najlepšia. Pomôžete mu ohodnotiť jednotlivé ponuky?

Ponuka o každej zastávke hovorí, ktorý deň sa má rekonštruovať. *Kvalita* ponuky je najmenšie číslo dňa, v ktorý sa na niektorej linke nebude rekonštruovať ani jedna zastávka.

### Úloha

Na vstupe dostanete jednu ponuku rekonštrukcie – pre každú zastávku dostanete deň, v ktorý sa podľa tejto ponuky má rekonštruovať. Ďalej dostanete zoznam všetkých liniek aj s ich konečnými zastávkami. Vypočítajte (a vypíšte) kvalitu tejto ponuky. Inými slovami, pre danú postupnosť dní a zoznam liniek vypíšte najmenšie číslo dňa, v ktorý sa na niektorej linke nerekonštruuje žiadna zastávka.

### Formát vstupu

V prvom riadku vstupu sú dve medzerou oddelené celé čísla  $n$  a  $m$  ( $1 \leq n, m \leq 1\,000\,000$ ) – počet zastávok a počet autobusových liniek.

Zastávky si očísľujeme číslami 1 až  $n$  od začiatku Kocúrkova po jeho koniec.

<sup>4</sup><https://www.ksp.sk/ulohy/zadania/1259/>

V druhom riadku vstupu je  $n$  medzerou oddelených čísel  $d_1, d_2, \dots, d_n$  ( $0 \leq d_i \leq 10^9$ ), kde  $d_i$  udáva číslo dňa od začiatku rekonštrukcie, kedy sa bude rekonštruovať zastávka číslo  $i$ . Rekonštrukcia začína dňom číslo 0.

Nasleduje  $m$  riadkov, v každom z nich sú dve medzerou oddelené čísla  $z_i$  a  $k_i$  – čísla konečných zastávok  $i$ -tej linky. Platí, že  $1 \leq z_i \leq k_i \leq n$ .

### Formát výstupu

Vypíšte jediné číslo – najmenšie číslo dňa, v ktorý sa na niektorej linke nerekonštruuje žiadna zastávka.

### Príklad

vstup	výstup
<pre>5 3 0 1 0 1 0 1 3 2 5 4 5</pre>	2

*V dni číslo 0 a 1 sa na každej linke niečo rekonštruuje. V deň číslo 2 sa však už nerekonštruuje nič, takže dokonca pre každú linku platí, že sa tam nič nerekonštruuje.*

vstup	výstup
<pre>5 4 0 1 2 1 1 1 3 2 3 1 4 3 5</pre>	0

*Na linke 2 3 sa nič nedeje v deň číslo 0.*

## 6. Oválna pracovňa

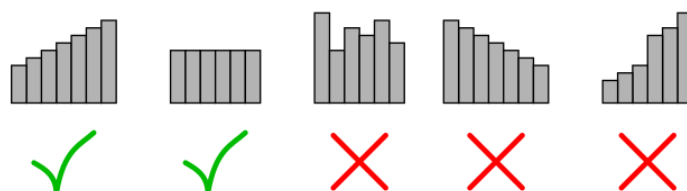
kat. O; 10 b za popis, 10 b za program

Nie je to tak dávno, čo sa do našej oválnej pracovne nasťahoval nový Týpek. Ako to už býva zvykom, nový týpek – nové nápady. A tak hneď ako ho napadol ten najlepší, zvolal svojich poradcov do oválnej pracovne. Za okrúhlym stolom im predostrel ideu postavenia veľkého múru pred nepriateľmi. Ako ale istá americká štúdia ukázala: čím bývajú nepriatelia viac na východ, tým sú nebezpečnejší. Preto musí byť múr na východe vyšší, alebo aspoň rovnako vysoký ako na západe.

Po nekonečných rokovaníach sa všetci v oválnej pracovni zhodli, že naša krajina je v ohrození a treba začať so stavbou ihneď. Ešteže zostalo v oválnej pracovni pod stolom zopár betónových stĺpov jednotkovej šírky. Stavbu teda môžeme zahájiť, no tak ako to pri veľkých projektoch býva potrebujeme plán, ten najlepší plán.

Hoci postavenie múru nie je vôbec populistické, zlé jazyky nás môžu začať ohovárať, preto musí mať náš múr aj nejaký hlboký zmysel, napríklad umelecký. Aby sme teda pred svetom nevyzerali ako barbari, musí sa postaviť múr esteticky, teda tak, aby rozdiel výšok medzi ľubovoľnými susednými betónovými stĺpmi bol rovnaký. Okrem toho je tu už spomínaný fakt, že nepriatelia na východe sú nebezpečnejší. Preto musia výšky jednotlivých stĺpov v múre tvoriť od západu na východ neklesajúcu postupnosť.

Na obrázku môžete vidieť päť rôznych múrov.



Prvé dva z nich sú dobré. Tretí je zlý, pretože výšky stĺpov netvoria neklesajúcu postupnosť. Štvrtý tak isto (tam stĺpy tvoria klesajúcu postupnosť). Piaty múr je zlý, lebo rozdiely medzi výškami susedných stĺpov nie sú všade rovnaké.

Pomôžte zachrániť krajinu pred čo najviac nepriateľmi a navrhňte čo najdlhší múr z materiálov, ktoré sú k dispozícii.

## Úloha

Máme  $n$  betónových stĺpov s danými výškami. Zistíte najväčšiu možnú dĺžku (počet použitých stĺpov) neklesajúceho múru ktorý vieme postaviť tak, aby v postavenom múre mali každé dva susedné betónové stĺpy rovnaký rozdiel výšok. Nemusíte použiť všetky stĺpy zo vstupu.

### Formát vstupu

Na prvom riadku vstupu dostanete kladné celé číslo  $n$ . V druhom riadku bude  $n$  medzerou oddelených nezáporných celých čísel  $a_1, a_2, \dots, a_n$  reprezentujúcich výšky jednotlivých stĺpov.

Je päť testovacích sád. Pre jednotlivé sady platia nasledovné obmedzenia:

číslo sady	1	2	3	4	5
$n \leq$	20	100	500	1 000	3 000
$a_i \leq$	$10^3$	$10^3$	$10^3$	$10^9$	$10^9$

### Formát výstupu

Na výstup vypíšte jediné číslo – počet použitých stĺpov v najdlhšom možnom múre spĺňajúcom podmienky zo zadania.

### Príklad

vstup	výstup
4 2 3 4 1	4

*Zjavne dokážeme postaviť múr zo všetkých našich stĺpov a tiež splniť estetickú podmienku, stačí postaviť stĺpy v poradí 1, 2, 3, 4*

vstup	výstup
5 2 1 5 2 4	2

*Najdlhší múr vieme postaviť napríklad zo stĺpov výšok 2 a 4*

## 7. Organizácia projektov

kat. O; 12 b za popis, 8 b za program

Syseľ pracuje ako manažér softvérových projektov. Práve teraz sa snaží vymyslieť, ako čo najoptimálnejšie priradiť programátorov ku dvom projektom. O každom z programátorov vie, ako je zbehlý v technológiách potrebných na ten-ktorý projekt. Pracuje s obmedzeným rozpočtom, preto si na každý z projektov môže dovoliť iba určitý počet programátorov, zvyšok žiaľ bude musieť prepustiť. Sysel by chcel nájsť čo najlepšie priradenie programátorov ku projektom. Pomôžete mu s týmto problémom?

## Úloha

Syseľ má  $n$  programátorov. Z jeho výpočtov mu vyšlo, že na projekte A môže pracovať  $x$  programátorov a na projekte B môže pracovať  $y$  programátorov. Zároveň pre každého programátora vie, aké veľké skúsenosti má s technológiami na projekte A a na projekte B. Hodnoty skúseností pre  $i$ -teho programátora si označíme  $a_i$  a  $b_i$ . Skúsenosť tímu, ktorý pracuje na projekte A je súčet hodnôt  $a_i$  všetkých programátorov pracujúcich na tomto projekte. Analogicky, skúsenosť tímu pracujúceho na projekte B je súčet hodnôt  $b_i$  všetkých programátorov, ktorí na ňom pracujú. Sysľovým cieľom je maximalizovať súčet skúseností oboch tímov, pričom jeden programátor môže pracovať iba na jednom projekte naraz. Povedané formálne, snažíme sa maximalizovať:

$$\sum_{i \in P_A} a_i + \sum_{i \in P_B} b_i,$$

pričom  $P_A$  je množina programátorov pracujúcich na projekte A a  $P_B$  je množina programátorov na projekte B.

### Formát vstupu

Na prvom riadku sa nachádzajú tri čísla  $n, x, y$  – celkový počet programátorov a počty programátorov, ktorí môžu pracovať na projekte A a na projekte B. Pritom platí:  $x + y \leq n, 2 \leq n \leq 10^5$  a  $x, y \geq 1$ .



Druhý riadok obsahuje čísla  $a_1, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), kde  $a_i$  je skúsenosť  $i$ -teho programátora s technológiami používanými na projekte A.

Tretí riadok obsahuje čísla  $b_1, \dots, b_n$  ( $1 \leq b_i \leq 10^9$ ), kde  $b_i$  je skúsenosť  $i$ -teho programátora s technológiami na projekte B.

### Formát výstupu

Na výstupe sa nachádza jedno celé číslo – maximálny možný súčet skúseností oboch tímov.

### Hodnotenie a obmedzenia

Pre jednotlivé sady testov navyše platia nasledovné obmedzenia. Za vyriešenie každej sady získate 2 body.

číslo sady	obmedzenie na $a_i, b_i$	obmedzenie na $n$
1	$1 \leq a_i, b_i \leq 1000$	$2 \leq n \leq 10$
2	$1 \leq a_i, b_i \leq 10^9$	$2 \leq n \leq 10^2$
3	$1 \leq a_i, b_i \leq 10^9$	$2 \leq n \leq 10^3$
4	$1 \leq a_i, b_i \leq 10^9$	$2 \leq n \leq 10^5$

### Príklady

vstup	výstup
5 2 2 1 3 4 5 2 5 3 2 1 4	18

*Sysel' priradí tretieho a štvrtého programátora na projekt A. Prvého a piateho priradí na projekt B. Druhého prepustí. Takto získa tím A s celkovou skúsenosťou  $4 + 5 = 9$  a tím B s celkovou skúsenosťou  $5 + 4 = 9$*

vstup	výstup
4 2 2 10 8 8 3 10 7 9 4	31

*Skúsenosť tímu A je  $10 + 8 = 18$  a tímu B je  $9 + 4 = 13$*

vstup	výstup
5 3 1 5 2 5 1 7 6 3 1 6 3	23

## 8. Oporné múry

kat. O; 12 b za popis, 8 b za program

Bude raz jeden dom. V tom dome bude bývať Samko. Keď ho postavia. A za tým domom bude kopec. Ten kopec tam dokonca je už teraz. A je to strmý kopec. A ako to už so strmými kopcami býva, vždy je pri ňom riziko zosuvu pôdy. Samkovi by sa, samozrejme, takýto zosuv nepáčil (keďže by mu mohol poškodiť jeho dom). Preto sa rozhodol (ešte pred stavbou svojho domu) postaviť na svahu niekoľko oporných múrov, ktoré svah spevnia a zosuvom zabránia.

Nechal si urobiť geodetický prieskum kopca, dlho nad ním hútal a nakoniec naplánoval niekoľko múrov. Každý z týchto múrov bude vodorovný – pôjde po vrstevnici. Napláňované múry môžu mať rôzne dĺžky (nemusia byť na celú šírku svahu) a môžu byť v rôznych výškach (na rôznych vrstevniciach).

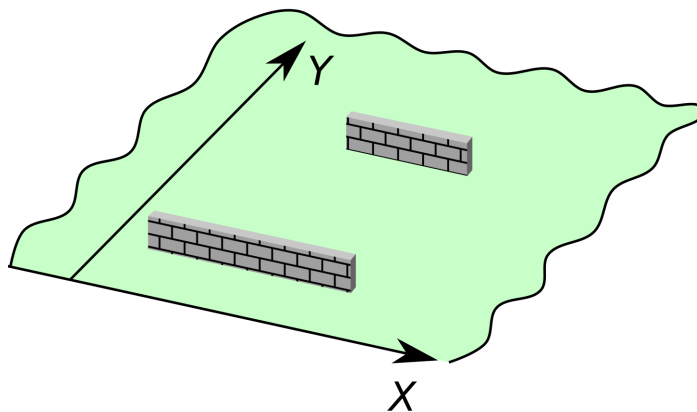
Plný entuziazmu si Samko naplánoval, v akom poradí jednotlivé múry postaví. Ešte v tom istom záchvate entuziazmu si tento plán nechal schváliť na stavebnom úrade. Neskôr, keď nadšenie trochu opadlo, si však uvedomil, že toto poradie možno nie je najšťastnejšie. Pri stavbe múru v strmom svahu sa vám totiž môže občas stať, že sa vám nejaký ten betónový kváder vymkne spod kontroly a skotúľa sa dolu kopcom. To je už samo o sebe trochu neprijemné, oveľa horšie však je, ak cestou narazí na nejaký iný, už postavený múr (snáď netreba vysvetľovať prečo).

Samko má už plán stavby schválený na úrade a nemôže sa len tak rozhodnúť, že múry postaví v inom poradí. Môžete mu však aspoň povedať, pri stavbe ktorých múrov hrozí, že by mu uvoľnený kus betónu poškodil nejaký nižšie položený, skôr postavený múr. Pri týchto múroch si potom bude na svoje betónové kvádre dávať špeciálny pozor.

## Úloha

Pre účely tejto úlohy si svah budeme predstavovať ako naklonenú rovinu (uhol naklonenia nie je podstatný). Na popisovanie miest na svahu si zavedieme nasledovnú súradnicovú sústavu: vrstevnica tvoriaca úpätie svahu bude  $x$ -ová os a jedna zo spádnic bude  $y$ -ová os, pričom  $y$ -ová súradnica bude rásť s rastúcou nadmorskou výškou. Múry teda budú v našom súradnicovom systéme zodpovedať úsečkám rovnobežným s  $x$ -ovou osou.

Hovoríme, že múr  $A$  ohrozuje múr  $B$ , ak je možné z nejakej časti múru  $A$  pustiť kameň tak, aby sa skotúlal a narazil do múru  $B$  (pričom kamene sa kotúľajú po spádniciach). Formálne, múr  $A$  ohrozuje múr  $B$ , ak je na vyššej  $y$ -ovej súradnici ako múr  $B$  a ich kolmé projekcie na os  $x$  majú prienik kladnej dĺžky (prienik v jednom bode teda nestačí).



Dostanete popis jednotlivých múrov a poradie, v akom ich Samko bude stavať. O múre  $C$  povieme, že je *riskantný*, ak existuje nejaký múr  $D$  taký, že  $C$  ohrozuje  $D$ , ale múr  $D$  bude postavený skôr než múr  $C$ . Pre každý múr rozhodnite, či je riskantný, alebo nie.

### Formát vstupu

Prvý riadok vstupu obsahuje jedno celé číslo  $n$  ( $1 \leq n \leq 100\,000$ ) – počet múrov. Nasleduje  $n$  riadkov, každý z nich popisuje jeden múr. Popis jedného múru sa skladá z troch medzerami oddelených celých čísel  $x_{i,1}$ ,  $x_{i,2}$ ,  $y_i$ , ktoré znamenajú, že daný múr je úsečka s koncovými bodmi  $(x_{i,1}, y_i)$  a  $(x_{i,2}, y_i)$ . Pri každom múre bude pre tieto čísla platiť  $-10^9 \leq x_{i,1} < x_{i,2} \leq 10^9$  a  $0 \leq y_i \leq 10^9$ . Navyše platí, že všetky múry sú navzájom disjunktné. Inými slovami, žiadne dva múry nemajú spoločný bod.

Múry si očísľujeme  $1, 2, \dots, n$  v poradí, v akom sú uvedené na vstupe. Posledný riadok vstupu obsahuje medzerami oddelené čísla múrov v poradí, v akom ich Samko bude stavať. Každé z čísel  $1, 2, \dots, n$  sa v poslednom riadku vstupu vyskytne práve raz.

### Formát výstupu

Pre každý múr (v poradí, ako sú očíslované) vypíšte jeden riadok obsahujúci slovo ANO ak je daný múr riskantný, resp. NIE, ak nie je riskantný.

### Príklady

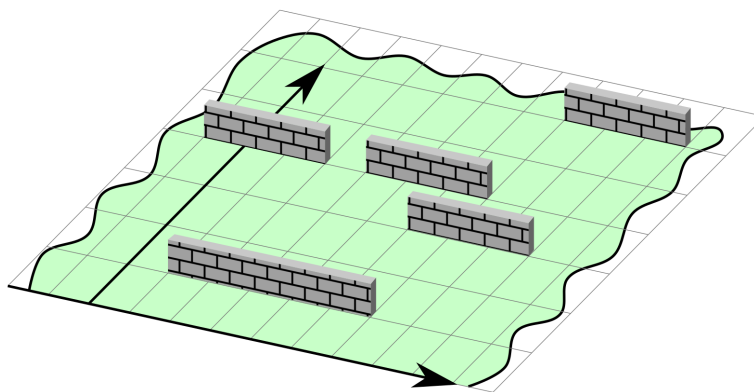
vstup

```
5
1 6 1
6 9 6
-1 2 4
5 8 3
3 6 4
5 1 3 2 4
```

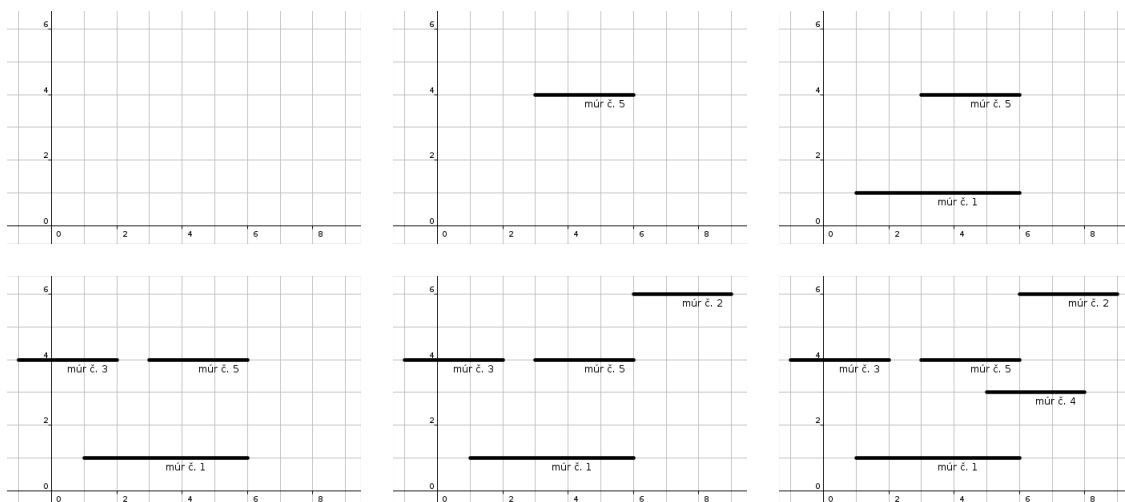
výstup

```
NIE
NIE
ANO
ANO
NIE
```

Po postavení všetkých múrov bude svah vyzerat nasledovne:



Múry budú postupne pribúdať takto:



Stavba múrov č. 5 a 1 bude bez rizika. Pri stavbe múru č. 3 hrozí poškodenie múru 1. Následná stavba múru č. 2 je opäť bez rizika. Nakoniec, pri stavbe múru 4 hrozí poškodenie múru 1. Riskantné sú teda múry č. 3 a 4.

## Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trochu ťažšia ako kategória O, ale mnohí z vás ju určite zvládnu).

Body z tejto kategórie sa mierne zohľadňujú aj pri výbere tímu, ktorý pôjde súťažiť na [Medzinárodnú olympiádu v informatike \(IOI\)](http://www.ioi.sk/), tento rok v Iráne.