



Leták zimnej časti VI. ročníka

Ahojte milí riešitelia.

Sme veľmi radi, že ste sa dozvedeli o PRASKu a asi by vás zaujímalo, čo to vlastne je, ako to celé funguje a prečo by ste to mali riešiť. Na všetko z toho sa vám teraz pokúsime odpovedať.

Čo to je a pre koho je to určené?

PRASK je korešpondenčný seminár určený pre všetkých základoškolákov, ktorých zaujíma matematika, informatika alebo by sa chceli naučiť programovať. Je to súťaž zameraná hlavne pre siedmakov a starších, môžete ju však riešiť aj keď ste v nižšom ročníku.

Seminár je organizovaný študentami informatiky na Fakulte matematiky, fyziky a informatiky na Univerzite Komenského.

Priebeh súťaže

Počas roka prebiehajú dve nezávislé časti – letná a zimná. Priebeh častí je už potom úplne rovnaký. Každá časť pozostáva z dvoch sérií piatich príkladov – dvoch teoretických, jedného praktického a dvoch programátorských. Ak aj neviete programovať nezúfajte. Namiesto programátorských úloh si môžete prejsť programátorským tutoriálom, ktorý vás to naučí a navyše v ňom získate body, ktoré sa vám rátajú do PRASKu.

Na riešenie série je vyhradených niekoľko týždňov. Až do dňa odovzdania môžete doma riešiť zadané príklady. Môžete riešiť ľubovoľné príklady z danej série, nemusíte vyriešiť všetko, nemusíte vyriešiť ani celú úlohu¹. Najneskôr do dňa odovzdania (ktorý je napísaný na zadaniach aktuálnej série) je potrebné poslať vaše riešenia pomocou webového rozhrania.

Po konci série si pozrieme vaše odovzdané riešenia a opravíme ich. Pre každý príklad je v zadaní napísané, koľko bodov sa zaň dá dostať. Samozrejme, je možné získať čiastkové body, aj keby ste nevyriešili celú úlohu, alebo by vaše riešenie nebolo úplne správne. Dokonca, ak nás prekvapíte originálnym riešením, môžete získať bonusové body. Opravené riešenie vám potom pošleme späť aj s poznámkami ohľadom vašeho riešenia.

Prečo to chceme riešiť?

Riešenie korešpondenčného seminára prináša mnoho výhod. Riešením úloh a čítaním našich vzorových riešení **objavíte a naučíte sa** mnoho nových vecí, ktoré by ste sa v škole skoro určite nenaučili. Napríklad sa môžete naučiť **programovať**. To vám potom vie **pomôcť pri prijímačkách**, či už na stredné alebo vysoké školy. Takisto vám to pomôže pri **riešení Olympiády z informatiky alebo Korešpondenčného Seminára z Programovania**. No a v neposlednom rade, pri **pohovoroch** do veľkých firiem ako Google, Facebook alebo Eset častokrát zaváži znalosť algoritmického programovania, ktoré si môžete pomocou nášho semináru trénovať.

Je tu však ešte jedna výhoda určená pre najlepších riešiteľov. Dvakrát ročne sa bude organizovať **týždenné sústredenie**. Naň pozývame niekoľko² najlepších riešiteľov. Na sústreďení si užiješ kopec zábavy, športu, nových ľudí a možno sa aj niečo naučíš.

A samozrejme, víťazov čakajú pekné **vecné ceny** vo forme knižky, hry alebo menšej elektroniky.

Ako má vyzeráť správne riešenie

To závisí od typu úlohy, ktorú riešite. Pri teoretických úlohách musí správne riešenie okrem výsledku obsahovať aj popis postupu, akým ste sa k danému výsledku dopracovali. Dôraz sa pri opravovaní dáva hlavne na tento slovný popis, ktorý by mal byť napísaný čo najzrozumiteľnejšie, aby sme si pri opravovaní nemuseli lámať hlavu. Mal by obsahovať všetky podstatné kroky, ktoré vás viedli k riešeniu.

V prípade praktických úloh sa to líši. Občas od vás chceme slovný popis, občas sa stačí dostať k nejakému tajnému heslu alebo kliknúť na správnu linku. Presný spôsob nájdete v zadaní.

¹Aj keď budeme radi, ak sa vám to podarí.

²zhruba 15, ale aj nižšie umiestnení riešitelia sa môžu dostať ako náhradníci

No a pri programátorských úlohách a programátorskej liahni odovzdávate iba váš program, ktorý sa vám okamžite automaticky otestuje a do pár sekúnd sa dozviete, či ste úlohu vyriešili správne. A ak nie, môžete skúsiť odovzdať opravený program znova.

A nebojte sa, ak ste ešte nikdy nespisovali postupy svojich riešení. Keď vám riešenia opravíme, napíšeme vám k nim aj komentáre, ktoré vám môžu pomôcť v riešení ďalšej série. To je najlepší spôsob, ako sa zlepšovať.

Spôsob odovzdávania

Ako prvú vec, ktorú musíte urobiť pred tým, ako budete môcť odovzdávať svoje riešenia, je **zaregistrovanie** sa na našej webovej stránke prask.ksp.sk. V časti **Zadania** a **vzoráky** nájdete okrem zadání aj odkaz, na ktorom môžete odovzdať vaše riešenie.

Riešenie každej teoretickej úlohy má byť jeden súbor formátu **.pdf**. Ten nahráte na našu stránku a stlačíte zelené tlačítko **Submit**. Opravovať sa bude **posledné odovzdané** riešenie, takže si dajte pozor, aby ste si niečo neprepísali.

Myslím, že vytvoriť pdf súbor by pre vás nemal byť problém, ak by ste s tým predsa len problém mali, pokúste sa použiť nejaký online converter ako napríklad www.freepdfconvert.com.

V prípade programátorských úloh sa dá rovnakým spôsobom odovzdať zdrojový kód vášho programu, teda súbor s príponou **.cpp**, **.py** alebo **.pas**.

Úlohy 2. kola zimnej časti

Termín odoslania riešení tejto série je pondelok 20. január 2020.

Teoretické úlohy

V tejto časti ťa čaká niekoľko matematickejších úloh, ktoré úzko súvisia s informatikou. Ako riešenie týchto úloh treba poslať podrobne spísaný postup toho, ako si riešil danú úlohu.

A ak by ťa to zaujímalo, podobné úlohy môžeš nájsť aj v Olympiáde v informatike, kategória B (<http://oi.sk>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

1. Práca v továrni dláždení

15 bodov za riešenie

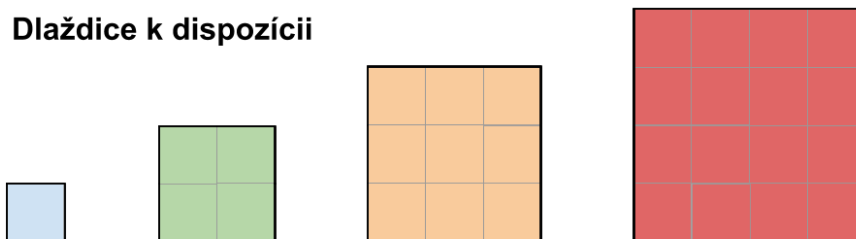
Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Michalovi Žabovi Anderlemu na zaba@ksp.sk

V školskom bufete opäť dvihli ceny a vreckové vám nestačí ani na kúpu vašej obľúbenej sladkej maškrtky. Nadišiel čas nájsť si brigádu! Pozreli ste si teda inzeráty v novinách a zaujala vás ponuka v “Továrni dláždení” – mzda slušná, podmienky prijateľné a riešenie PRASKu výhodou. Po krátkom pohovore ste boli razom prijatí.

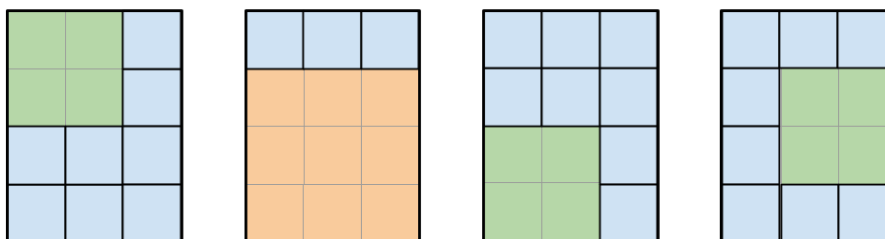
Postavili ste sa k pásu, poblíž boli štyri škatule, v každej bolo **neobmedzene** veľa dlaždíc veľkostí 1×1 , 2×2 , 3×3 a 4×4 . Asi ich budete baliť do menších zásielok. . . Po páse k vám však prišla iba obálka s listom, na ktorom bola napísaná nasledovná otázka: “Koľko existuje rôznych dláždení plochy 4×3 pomocou dlaždíc 1×1 , 2×2 , 3×3 a 4×4 ?”

Ostali ste zaskočení, pokrčili ste však plecami nad touto zvláštnosťou³ a pustili ste sa do práce. Rýchlo ste zistili, že hľadanou odpoveďou je 13. Na obrázku nižšie si môžete pozrieť štyri z týchto možností, zvyšné si môžete nakresliť sami. Na list ste teda napísali odpoveď, poslali ste ho ďalej a čakali na ďalšiu požiadavku.

Dlaždice k dispozícii



4 možné dláždenia plochy 4x3



Po prvom dni vám bolo jasné, že cieľom továrne je odpovedať na otázky o počte rôznych dláždení plochy $4 \times n$, výška dotazovanej plochy je **vždy 4**, pomocou spomenutých štyroch typov dlaždíc.

Úloha

³Kto potrebuje odpoveď na túto otázku?

- a) (2 body) Vypočítajte, koľko existuje rôznych možností dláždenia plochy 4×4 . Skúste stručne popísať, ako ste pri počítaní postupovali a prečo ste nevynechali žiadnu možnosť.

Pomerne rýchlo váš však táto práca omrzela a začali ste rozmýšľať, ako si ju uľahčiť. A veru ste vymysleli výborný trik. Jednoducho sa spýtate svojich kolegov! Samozrejme, nemôžete to spraviť priamo, lebo by vám určite nepomohli. Stačí ale, že spravíte list s požiadavkou a zamiešate ju medzi doručeníu poštu. Potom si počkáte na odpoveď a bude. Má to však jeden háčik. Ak sa spýtate tú istú otázku ako prišla vám, bude to pôsobiť podozrivo. Pýtať sa teda môžete iba na menšie problémy.

Na riešenie zvyšných podúloh môžete využívať nasledovnú operáciu. Ak máte zistiť počet možností pre plochu $4 \times n$, môžete sa spýtať ľubovoľne veľa otázok, ktoré sú typu “Koľko existuje rôznych dláždení plochy $4 \times m$?”, pričom musí platiť, že m je menšie ako n . Následne predpokladajte, že na tieto otázky vám prišla správna odpoveď a takto získané odpovede môžete použiť na riešenie svojho problému.

- b) (2 body) Dostali ste mierne upravenú otázku: “Koľko existuje rôznych dláždení plochy $4 \times n$ takých, že v poslednom stĺpci sa nachádzajú iba dlaždice 1×1 ?”. Popíšte spôsob, akým zistíte odpoveď na túto otázku s využitím vyššie uvedeného triku.
- c) (4 body) Popíšte postup, akým zistíte odpovede na nasledovné tri otázky:
- Koľko existuje rôznych dláždení plochy $4 \times n$ takých, že v poslednom stĺpci je aspoň jedna dlaždica 2×2 ?
 - Koľko existuje rôznych dláždení plochy $4 \times n$ takých, že v poslednom stĺpci je aspoň jedna dlaždica 3×3 ?
 - Koľko existuje rôznych dláždení plochy $4 \times n$ takých, že v poslednom stĺpci je aspoň jedna dlaždica 4×4 ?

- d) (2 body) Popíšte postup, ktorý použijete na odpovedanie otázky “Koľko existuje rôznych dláždení plochy $4 \times n$?”.

- e) (3 body) Po továrni sa rozniesol váš postup z podúlohy d), ktorý využívate na zjednodušenie práce. A samozrejme ho začali používať aj všetci vaši kolegovia. Väčšina správ v továrni je teraz od jej vlastných zamestancov. Vašou najnovšou úlohou je zistiť, koľko existuje dláždení plochy 4×7 . Nakreslite, ako bude prebiehať zisťovanie odpovede na túto otázku. Zaznačte všetky správy, ktoré bude v továrni poslané pre jej vyriešenie.

- f) (2 body) Pozorne si prezrite svoj obrázok z podúlohy e). Je takýto postup efektívny? Robí sa tam niečo navyše a ak áno, ako by ste to vedeli zlepšiť.

V továrni je okrem iného aj obrovská korková tabuľa, na ktorú sa pripínajú rôzne odkazy a momentálne je prázdna. S kolegami by ste si chceli prácu ešte viac uľahčiť. Skúste navrhnúť spôsob akým môže fungovať odpovedanie na nové otázky tak, aby nikto z vás nerobil zbytočnú robotu.

2. Rizikové Receptomaty

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Adamovi Královi na adam.kral@trojsten.sk

Táto úloha nadväzuje na úlohu [Receptomaty](#) z prechádzajúceho kola. Pri riešení tejto úlohy preto môžete využívať vzorové riešenie z minula.

Čarodejnice(a čarodejnici) vám to síce nechceli povedať, ale kniha receptomatov má druhú časť s názvom “Nerozhodné receptomaty”. Nechceli vám to povedať, pretože týmto receptomatom nerozumeli. Šípky zo stavov nedávajú veľmi zmysel - niekedy je ich viac ako jedna, čiže čarodejníci sa nevedeli dohodnúť, po ktorej majú ísť. Nakoniec však rozlúštili túto záhadu a popis k týmto receptomatom vám pridali do tutoriálu. Ten vám zase oskenovali a zdieľali na [tejto adrese](#). Aby vám pomohli ešte viac, updatovali aj simulátor receptomatu. Ten je na [tejto adrese](#).

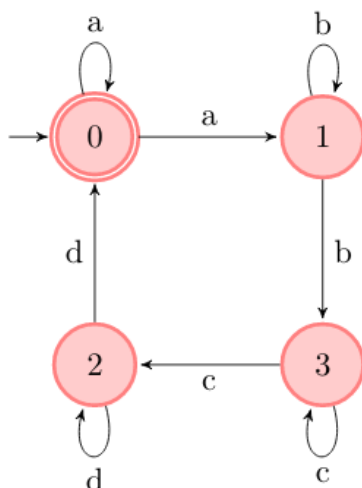
Zároveň potrebujú od vás znovu pomoc.

Úloha

Niektoré strany z knihy receptomatov chýbajú. Pomôžte nájsť:

- a) (2 body) Nerozhodný receptomat, ktorý akceptuje reťazce s podreťazcom **abac**. Poznámka: táto úloha je rovnaká ako v prvej časti, typ receptomatu však výrazne zjednodušuje riešenie.

- b) (2 body) Receptomat, ktorý akceptuje čísla deliteľné číslom 11. (môže byť nerozhodný alebo rozhodný)
- c) (2 body) Nerozhodný receptomat, ktorý akceptuje reťazce končiace sa na aa.
- d) (2 body) Receptomat z c, ale aby bol rozhodný.
- e) (2 body) Nerozhodný receptomat, ktorý akceptuje čísla deliteľné číslom 25.
- f) (2 body) Receptomat z e, ale aby bol rozhodný.
- g) (3 body) Aké recepty akceptuje tento receptomat? Dokážete z neho spraviť rozhodný receptomat?



Podúlohy *a*, *c*, *d* sú z kapitoly “Písmenné receptomaty”. V týchto receptoch sa používajú len znaky *a*, *b*, *c* a *d*. Podúlohy *b*, *e*, *f* sú z kapitoly “Číselné receptomaty”. V týchto sa používajú znaky 0 až 9.

Praktická úloha

Pri práci s počítačom je potrebné vedieť pracovať aj s rôznymi nástrojmi, ktoré slúžia na úpravu obrázkov, prácu so zvukom či vyhľadávaním na internete. V tejto časti ťa preto zakaždým čaká nejaká netradičná úloha.

3. A tak sa Prefix stratil

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Prefixovi na michalsladecek98@gmail.com

Táto úloha nadväzuje na úlohu [A tak sa Denis stratil](#) z prechádzajúceho kola. Pri riešení tejto úlohy preto môžete využívať vzorové riešenie z minula.

Podarilo sa vám dostať k poslednému kódu. Zapísali ste ho do súboru spolu s ostatnými kódmi. Zrazu ste začuli hlasné *vžuuu* a Denis sa z čista jasna vynoril z počítača. Predsa len sa vám podarilo zachrániť ho. Denis vám celý nadšený poďakoval a rýchlo bežal⁴ po Prefixa. Niečo takéto mu predsa musí ukázať. Prefix je totižto skúsený hackerman a tento program ho bude isto zaujímať.

Denis postupne porozprával celý príbeh Prefixovi. Prefix, vzhľadom na svoju expertízu, Denisovi neveril. Ako by sa predsa mohol niekto stratiť v príkazovom riadku? Nemožné. Rozhodol sa, že sa na daný program musí pozrieť osobne. Predsa len, taký skúsený hackerman ako on rozlúskne, čo sa stalo. Veď je výrazne skúsenejší a nespraví rovnaké chyby ako Denis. Prefix príde k počítaču a spustí daný program ako správca. *Vžuuuuuuuuuu.*

Vašou úlohou je tentokrát zachrániť Prefixa. Keďže Prefix je skúsený hackerman, príkazový riadok si ho bude brániť výrazne viac ako Denisa. Prefix je v počítači roztrúsený v podobe kódov, ktoré sa nachádzajú v jednotlivých úlohách. Každý kód vyzerá ako spleť znakov a čísel, napr. `d003f58b96e6b0a543bf75`.

“Ale ja sa neviem orientovať v príkazovom riadku, ako ho tam mám nájsť?” hovoríš si. Žiaden problém! Všetko potrebné na začiatok nájdeš v [predchádzajúcej úlohe](#).

⁴Autor navádza na akútnosť rozhodnutia, keďže Denis len tak pre nič za nič nebeží, a nikdy nebeží rýchlo.

“To som si už predsa pozrel, táto úloha sa mi ale zdá ťažšia. Ako ho mám teda nájsť?”. Opäť raz si budeš musieť veľa vecí vygoogliť, no čo by si pre zmenu nespravil/a pre Prefixa?

Úloha

V tejto úlohe budete musieť vykonávať, podobne ako v úlohe, kde ste zachraňovali Denisa, rôzne podúlohy v príkazovom riadku Linuxu. Tento systém nebude bežať u vás – dostanete sa doň pomocou [návodu](#), ktorý sme pre vás spísali. Takže sa nebojte, nemusíte si nainštalovať Linux.

V prípade, že si neviete rady, Google je váš kamarát. Pokiaľ viete, môžete skúsiť hľadať po anglicky. Väčšinou dostanete viac výsledkov, ktoré budú často aj presnejšie.

Či ste vyriešili príklad zistíte tak, že sa dostanete ku kódu, ktorý môže vyzeráť napríklad takto `d003f58b96e6b0a543bf75`. Tieto kódy budete na konci odovzdávať v popise. Požadujeme od vás aj **krátky popis** toho, ako ste príklad riešili. Nepripravili sme pre vás žiadne chytáky, takže keď už sa v príklade dostanete k niečomu, čo vyzerá ako kód, skoro určite je správny a dostanete bod.

Programátorské úlohy

Tieto úlohy sú zamerané na praktickú tvorbu programov v niektorom vyššom programovacom jazyku ako je napríklad Python, C++ alebo Pascal. Na stránke odovzdávaš **iba zdrojový kód** svojho programu riešiaceho zadanú úlohu, ktorý bude okamžite automaticky otestovaný a do pár sekúnd sa dozvieš, koľko bodov tvoj program získal. Tieto body ti už nikto nemôže zobrať, ale ak si nezískal plný počet bodov, môžeš opakovane odovzdávať opravený program, až kým nebudeš spokojný s výsledkom.

Ak už vieš programovať, ale ešte si nepracoval s naším testovacím systémom, odporúčam ti zájsť na Programátorskú Liaheň (<http://liahen.ksp.sk>), kde si o tom môžeš prečítať úvodný text a vyriešiť si niekoľko jednoduchých úloh.

Ak však **nevieš programovať, tak nezúfaj!** Pripravili sme pre teba **Programátorskú Liaheň**, ktorá ťa **naučí základy programovania** v jazyku C++. Navyše, za riešenie týchto tutoriálových úloh na Liahni môžeš získať body priamo do PRASKu a tým si vynahradiť neriešenie niektorej z programátorských úloh.

Presnejšie to funguje takto. Na Liahni sa nachádzajú dve sady úloh, prvá zameraná na premenné a druhá na podmienky v jazyku C++. V týchto sadách sa nachádzajú bodované aj nebodované úlohy, ktoré môžeš postupne riešiť a ktoré ti postupne vysvetlia danú problematiku. Dokopy sa v jednej sade dá získať až 15 bodov.

Týmito bodmi si potom môžeš nahradiť úlohy 4 a 5. Samozrejme, toto môžeš urobiť **s každou sadou najviac raz**.

No a v budúcej sérii budeš môcť za body riešiť ďalšie dve sady z Liahne.

Samozrejme, nič ti nebráni riešiť aj úlohy z Liahne aj klasické programátorské úlohy v PRASKu.

Programátorskú Liaheň nájdeš na tejto stránke: <http://liahen.ksp.sk>

4. Spájanie rebríkov

15 bodov za riešenie

Ak nevieš programovať, nezúfaj! Môžeš sa to naučiť a ešte za to získať body, ktoré sa ti budú počítať namiesto tejto úlohy.

Stačí, že pôjdeš na stránku Programátorskej Liahne (liahen.ksp.sk). Keď budeš riešiť sadu **loops_cpp**, bodmi, ktoré získaš, si môžeš nahradiť riešenie tejto úlohy. Stačí ak na spodku tejto stránky odovzdaš pdf-ko s prezývkou, ktorú používaš na Liahni.

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Martinovi na martin@kalab.sk

Kde bolo tam bolo, išli si Maťko s Kubkom hádzať frisbee. Keďže už bol večer, ich starostlivá mamička im nedovolila ísť ďalej ako pred ich panelák. Pri hádzaní skúšali chalani rôzne triky a blbosti. Zrazu sa Maťkovi šmykla ruka a zhrozene pozeral, ako frisbee letí a pristáva na balkóne jedného z ich susedov.

Našťastie ešte nie je nič stratené. Ich dedo Jozef má totiž veľkú zbierku skladacích rebríkov, od úplne maličkých až po rebríky vyššie ako hora. Ale nie sú to len také hocikaké rebríky. Každý rebrík má špeciálnu prípojku, pomocou ktorej sa dokáže spojiť s ľubovoľným iným rebríkom. Dedo Jozef je taktiež veľmi poriadkumilovný, takže svoje rebríky má usporiadané podľa veľkosti.

Maťko a Kubko sa teda rozhodli, že sa spýtajú deda, či by im pár rebríkov nepožičal. Dedo Jozef súhlasil, ale dovolil im spojiť najviac **dva** rebríky. Ak by ich totiž spojili viac, spojené rebríky by mohli byť veľmi labilné. Výsledný rebrík nesmie byť primalý, lebo na frisbee nedosiahnu, ale ani príliš vysoký, pretože im bude zavádzať a k frisbee sa nedostanú. Mal by byť tak akurát, presne po ten nešťastný balkón.

Dedo Jozef požičal Kubkovi a Mačkovi svoje rebríky, ale tí teraz potrebujú zistiť, či sa s nimi dostanú na balkón alebo nie.

Úloha

Vašou úlohou je napísať program, ktorý podľa dĺžok rebríkov a výšky balkóna zistí, či existuje **jeden** rebrík s výškou balkóna alebo či sa dá takýto rebrík zložiť z **dvoch** iných. Veľkosti rebríkov nemusia byť unikátne, čo znamená, že na vstupe sa môžu vyskytnúť dva rebríky rovnakej dĺžky.

Formát vstupu

Na prvom riadku sa nachádzajú dve medzerou oddelené čísla k a n ($1 \leq k \leq 10^9$, $1 \leq n \leq 300\,000$), prvé z nich označuje výšku balkóna, na ktorý spadlo frisbee, a druhé počet dedových rebríkov.

Na nasledujúcom riadku je n čísel, ktoré udávajú výšky rebríkov. Tieto čísla sú zoradené od najmenšieho po najväčšie a ich veľkosti sú medzi 1 a q ($1 \leq q \leq 10^9$).

Formát výstupu

Vypíšte na jediný riadok výstupu **ano**, ak sa z najviac dvoch rebríkov dá poskladať rebrík s výškou balkóna, inak **nie**.

Hodnotenie

Pripravili sme 3 sady vstupov. Platia pre ne nasledové horné obmedzenia pre hodnoty n , k a q .

Číslo sady	1	2	3
maximálne n	1000	300 000	300 000
maximálne k a q	1000	1 000 000	10^9

Príklady

vstup

```
6 5
1 2 3 3 4
```

výstup

```
ano
```

Výšku 6 vedia Mačko s Kubkom dosiahnuť buď pomocou dvoch rebríkov dĺžky 3 alebo rebríkov 2 a 4.

vstup

```
6 2
2 3
```

výstup

```
nie
```

Výška 6 by sa dala dosiahnuť napríklad dvoma rebríkmi dĺžky 3, ale k dispozícii máme iba jeden. Preto je odpoveď **nie**.

vstup

```
8 3
1 8 9
```

výstup

```
ano
```

Rebrík výšky 8 sa nachádza priamo medzi rebríkmi zo vstupu, takže odpoveď je **ano**.

5. Kvalita výberu

15 bodov za riešenie

Ak nevieš programovať, nezúfaj! Môžeš sa to naučiť a ešte za to získať body, ktoré sa ti budú počítať namiesto tejto úlohy.

Stačí, že pôjdeš na stránku Programátorskej Liahne (liahen.ksp.sk). Keď budeš riešiť sadu **conditions_cpp**, bodmi, ktoré získaš si môžeš nahradiť riešenie tejto úlohy. Stačí ak na spodku tejto stránky odovzdaš pdf-ko s prezývkou, ktorú používaš na Liahni.

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Danovi na danza@ksp.sk

Farmár Fero nie je farmár. Ale chcel by byť. Na to však, samozrejme, potrebuje pole. Blízko jeho domu je veľký kus hospodárskej pôdy, ktorú nikto nevyužíva. Je ako stvorená pre neho. Keď ale Fero zistil tú strašnú cenu za každý meter štvorcový tejto pôdy, uvedomil si, že si môže dovoliť odkúpiť iba nejaký menší kúsok.

Celá hospodárska pôda má štvorcový tvar. Fero by z nej chcel nejaký obdĺžnik. Prečo zrovna obdĺžnik? Keby to bol nejaký krivoľaký útvar, ťažko by sa mu na ňom manévrovalo s kombajnom. A on na tom poli predsa nič iné robiť nebude.

Hospodárska pôda je naozaj rozsiahla. Je to štvorec rozmerov $n \times n$ metrov. Niektoré jej časti sú úrodnejšie, ako iné. Terajší vlastník má pre každé políčko veľkosti 1×1 meter spočítané, ako veľmi je úrodné. Na niektorých políčkach neprežije ani burina, na iných vie vyrásť aj kukurica. Nájdu sa dokonca aj tak neúrodné políčka, že im terajší vlastník priradil zápornú hodnotu. Všetky tieto dáta poskytol Ferovi. Vraj aby sa mohol dobre rozhodnúť.

Náš odhodlaný takmer-farmár začal hútať, ktorý obdĺžnik by si mal vybrať, aby bol čo najúrodnejší. Vytipoval si niekoľko kandidátov. Teraz ale pre každý z vytipovaných obdĺžnikov potrebuje vedieť, aká je jeho úrodnosť. Pomôžete mu vybrať jeho pole?

Úloha

Kus hospodárskej pôdy má rozmery $n \times n$ metrov. Je tvorený políčkami rozmerov 1×1 meter. Ide teda o akúsi mriežku s n riadkami a n stĺpcami. Pre každé políčko poznáte jeho úrodnosť vyjadrenú celým číslom. Fero sa vás opýta q otázok. Každá otázka predstavuje obdĺžnik na veľkom kuse hospodárskej pôdy. Pre každý takýto obdĺžnik je vašou úlohou spočítať súčet úrodností jeho políčok.

Formát vstupu

Na prvom riadku vstupu je číslo n . To hovorí, že kus hospodárskej pôdy, z ktorej si Fero ide vybrať obdĺžnikové pole, má rozmery $n \times n$ políčok. Platí, že $1 \leq n \leq 2000$.

Nasleduje n riadkov. V každom z nich je n čísel. Číslo x_{ij} v i -tom riadku a j -tom stĺpci hovorí, že políčko v i -tom riadku a j -tom stĺpci hospodárskej pôdy má úrodnosť x_{ij} . Platí, že $-1000 \leq x_{ij} \leq 1000$.

Na ďalšom riadku je číslo q – počet otázok. Môžete predpokladať, že $1 \leq q \leq 100\,000$.

Nasleduje q riadkov – jednotlivé otázky. Každý riadok obsahuje 4 čísla r_1, c_1, r_2, c_2 . Odpoveďou na túto otázku je úrodnosť obdĺžniku, ktorý má ľavý horný roh v riadku r_1 a stĺpci c_1 , a pravý dolný roh v riadku r_2 a stĺpci c_2 . Platí, že $0 \leq r_1, c_1, r_2, c_2 \leq n - 1$ a tiež, že $r_1 \leq r_2$ a $c_1 \leq c_2$.

Formát výstupu

Pre každú otázku vypíšte jej odpoveď na jeden riadok. Odpovede vypisujte v takom poradí, v akom ste dostali otázky. To znamená, že v prvom riadku bude odpoveď na prvú otázku, v druhom na druhú, atď.

Hodnotenie

Sú 3 sady vstupov. Platia pre ne nasledové horné obmedzenia pre hodnotu q .

Číslo sady	1	2	3
$q \leq$	10	10 000	100 000

Príklad

vstup

```
4
1 -2 -3 4
-2 4 -1 2
1 -2 3 5
0 -1 9 9
2
1 1 2 3
1 3 1 3
```

výstup

```
11
2
```

Prvý obdĺžnik, na ktorý sa Fero pýta, je na obrázku vyplnený zelenou farbou. Druhý je zarámovaný červenou čiarou.

	0	1	2	3
0	1	-2	-3	4
1	-2	4	-1	2
2	1	-2	3	5
3	0	-1	9	9