



Leták zimnej časti III. 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 základuškolákov, 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 rátaajú 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 chcem 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 algoritmickeho programovania, ktoré si môžete pomocou nášho semináru trénovať.

Je tu však ešte jedna výhoda určená pre najlepších riešiteľov. Dvakrát ročne sa bude organizovať **týždenné sústreďenie**. 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 knihy, 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.

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

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

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.

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 zadaní 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 1. kola zimnej časti

Termín odoslania riešení tejto série je pondelok **22. máj 2017.**

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/archiv/2016/sl-2016-1-zad-B.pdf>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

1. Prehra na šachovnici

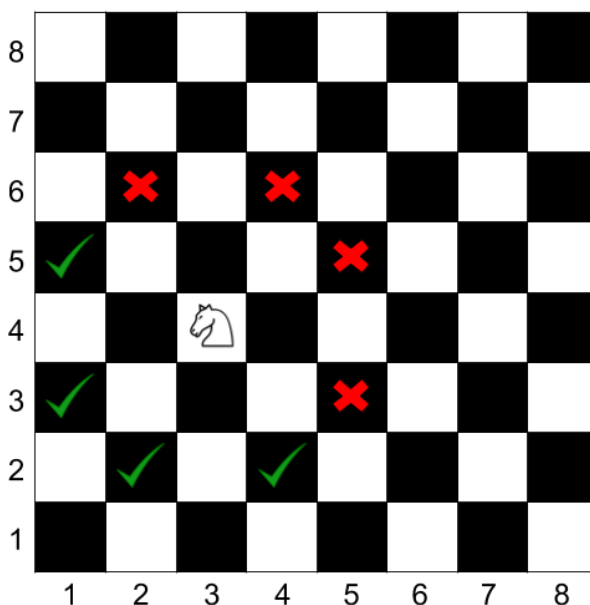
15 bodov za riešenie

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

Kika s Andrejom išli spolu na výlet do Tatier. Prenajali si malú chatku uprostred hôr, z ktorej sa každý deň chystali chodiť na rôzne horské túry. Avšak hneď v prvý deň ráno začalo strašne pršať a nikam ísť nemohli. Ostali teda uväznení na chate, a keďže tam nebol žiaden signál, začali sa strašne nudiť.

Snažili sa teda nájsť čokoľvek, čím by sa mohli zabaviť. Ale jediná vec, ktorú našli, bola stará šachovnica s figúrkami. Ani jeden z nich však nevie hrať šach, akurát vedia ako sa hýbu jednotlivé figúrky, preto si vymysleli nasledovnú hru.

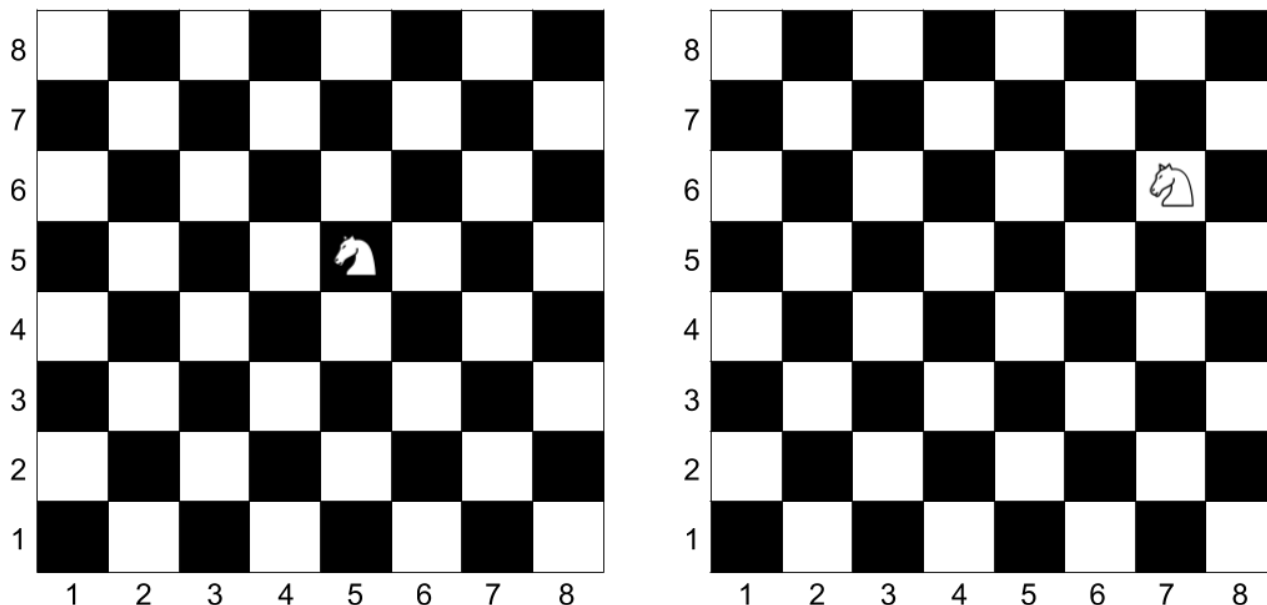
Na niektoré políčko šachovnice položia figúrku koňa. Následne sa striedajú v ťahoch – ťah spočíva v tom, že hráč presunie koňa podľa šachových pravidiel na iné políčko šachovnice. Navyiac musí platiť, že v každom ťahu sa koň posunie bližšie k ľavému dolnému rohu šachovnice. Formálne, očísľujme si riadky šachovnice odspodu po vrch číslami 1 až 8 a stĺpce šachovnice zľava doprava číslami 1 až 8. Potom musí platiť, že každým ťahom sa súčet súradníc, na ktorých stojí koň **zmenší**. Na nasledujúcom obrázku si môžete pozrieť všetky možné ťahy, ktoré môže šachový koň spraviť – koň sa pohybuje v L-ku, ktorého ramená sú dĺžky 3 a 1. Štyri z týchto ťahov sú prípustné (označené zelenou fajkou), lebo zmenšujú súčet súradníc a štyri z nich sú nepovolené (označené červeným krížikom). Samozrejme, ak sa koň nachádza na kraji šachovnice, niektoré z ťahov hráč nemôže spraviť, lebo by dostal koňa mimo šachovnicu.



Hráč, ktorý už koňom nemôže pohnúť (lebo sa nachádza v niektorom zo 4 políčok v ľavom dolnom rohu) prehráva. No a keďže Andrej je gentleman, Kika začína ako prvá.

Úloha

- a) (3 body) Na obrázkoch nižšie sú zobrazené dve možné začiatkové pozície. Pre každú z nich rozhodnite, ktorý hráč vie zvíťaziť, ak budú hrať obaja optimálne. Pýtame sa teda, ktorý z hráčov vie vyhrať bez ohľadu na to, aké ťahy spraví jeho protivník. Pre obe situácie popíšte aj spôsob, akým vie daný hráč vyhrať.



- b) (7 bodov) Pre každú pozíciu šachovnice 8×8 zistite, či sa podarí Kike zvíťaziť, ak bude kôň na začiatku na tejto pozícii. Výsledok zapíšte do tabuľky 8×8 , kde na pozíciu, z ktorej Kika vie vyhrať zapíšete číslo 1 a na pozíciu, z ktorej Kika prehrá zapíšete 0. Teda štyri políčka v ľavom dolnom rohu budú obsahovať 0, pretože ak sa na ne položí kôň, Kika nebude môcť spraviť žiaden ťah a prehrá.

Popíšte, akým spôsobom ste túto tabuľku vyplňali a zdôvodnite, prečo je váš postup naozaj dobrý. Potom sa pokúste zovšeobecniť váš postup aj pre šachovnice ľubovoľných rozmerov – teda veľkosti $r \times s$. Skúste preto popísať sadu pravidiel, podľa ktorých by ste vedeli vyplniť ľubovoľne veľkú tabuľku.

- c) (5 bodov) Keď ich táto verzia prestala baviť, hru si ozvláštnili tým, že na šachovnicu položili viacero koní (kludne aj viac ako štyroch). V jednom ťahu si mohol hráč vybrať ľubovoľne veľa koní (ale aspoň jedného) a každého z nich posunúť podľa predošlých pravidiel. Nevadilo dokonca ani to, ak sa viacero koní ocitlo na tom istom políčku.

Pomôžte Kike nájsť spôsob akým môže vyhrať, poprípade zistite, že vyhrať nevie. Napíšte teda postup, ktorým sa má v každom kroku riadiť. Tento postup musí byť dosť všeobecný, aby sa dal použiť pri ľubovoľnej situácii, ktorá na šachovnici nastane. Pri riešení môžete použiť tabuľku z podúlohy b).

2. Pestovanie termosenzitívneho ovocia v Absurdistane

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Mariánovi Horňákovi na sysel@ksp.sk

V predchádzajúcej sérii ste sa stretli s genetickými inžiniermi z Absurdistanu, ktorým ste pomohli navrhnúť rastliny produkujúce presné množstvá plodov každý rok. Projekt mal skvelý marketing a získal tisíce podporovateľov, ktorí si na niekoľko rokov dopredu objednali svoje ovocie. Slávou omámení inžinieri sa teda rozhodli pustiť do ešte ambicioznejšieho problému. Do presného a rýchleho plánovania úrody termosenzitívneho ovocia.

Úroda termosenzitívneho ovocia závisí od toho, či bol konkrétny rok studený alebo teplý. Pre obe situácie je presne určený počet aj typ plodov, ktoré na rastline vyrastú. Vedci dokážu ovládať počasie natoľko, aby vynútili konkrétne správne rastlín v každom roku. Teraz potrebujú zistiť, ako sa majú teplé a studené roky striedať tak, aby čo najskôr dosiahli úrodu konkrétnej veľkosti. Pomôžete im?

Úloha

Táto úloha nadväzuje na úlohu *Pestovanie ovocia v Absurdistane* z prechádzajúceho kola. Pred tým, než budete pokračovať v čítaní, pozrite si zadanie a prípadne aj vzorové riešenie minulej úlohy.

Rastliny budeme v tejto úlohe označovať veľkými písmenami anglickej abecedy. Každá rastlina bude mať určené, aké plody na nej dozrievajú počas teplého roka a aké počas studeného roka. Toto zapíšeme tak, že za písmeno, ktoré zodpovedá rastline, napíšeme šípku a za tú písmená tých rastlín, ktorých plody na nej dozrejú. Nad šípkou uvedieme T alebo S , podľa toho, či je to pravidlo pre teplý alebo pre studený rok. Ak na rastline nič nevyrastie, zapíšeme to ako $\rightarrow 0$. Kompletná sada pravidiel potom môže vyzeráť nasledovne:

$$\begin{array}{cc} \hline A \xrightarrow{T} B & A \xrightarrow{S} 0 \\ B \xrightarrow{T} AB & B \xrightarrow{S} B \\ \hline \end{array}$$

Každý typ rastliny musí mať priradené práve jedno pravidlo pre teplý rok a práve jedno pravidlo pre studený rok. Na poradí písmen nezáleží, teda $B \xrightarrow{T} AB$ je totožné s $B \xrightarrow{T} BA$.

V každom roku zasadíme všetky semená, ktoré máme. Podľa teploty, ktorú si zvolíme, potom vyrastie rastlina aj s plodmi. Každý plod obsahuje práve jedno semeno, ktoré zasadíme v ďalšom roku. Ak by sme sa teda riadili podľa vyššie uvedeného príkladu, mohla by sa úroda vyvíjať ktoroukoľvek z nasledovných možností:

$$\begin{array}{c} A \xrightarrow{T} B \xrightarrow{S} B \xrightarrow{T} AB \xrightarrow{T} BAB \xrightarrow{T} ABBAB \xrightarrow{S} BBB \xrightarrow{T} ABABAB \\ A \xrightarrow{T} B \xrightarrow{T} AB \xrightarrow{T} BAB \xrightarrow{T} ABBAB \xrightarrow{T} BABABBAB \xrightarrow{S} BBBBB \\ A \xrightarrow{S} 0 \xrightarrow{T} 0 \xrightarrow{S} 0 \end{array}$$

Postupne si teraz predstavíme tri sady pravidiel, ukážeme si príklad vývoja úrody a položíme niekoľko otázok o tom, akú veľkú úrodu možno zostrojiť.

Ananásová sada

Ananásová sada je pomerne jednoduchá. Začíname s ananásom (A), na ktorom okrem malého ananásu vyrastú aj nejaké bobule. Tie sa však už ďalej nemnožia, keďže z bobule vyrastie práve jedna bobuľa bez ohľadu na teplotu. Pravidlá sú nasledovné:

$$\begin{array}{cc} \hline A \xrightarrow{T} AB BB & A \xrightarrow{S} AB \\ B \xrightarrow{T} B & B \xrightarrow{S} B \\ \hline \end{array}$$

Ak by sme chceli dosiahnuť 9 plodov, mohli by sme to spraviť napríklad takto:

$$A \xrightarrow{T} AB BB \xrightarrow{S} AB BB B \xrightarrow{T} AB BB BB BB \xrightarrow{S} AB BB BB BB B$$

- (1 bod) Nájdite takú postupnosť teplých a studených rokov, že úroda po posledom roku bude obsahovať presne 42 plodov.
- (2 body) Po koľkých rokoch najskôr a po koľkých rokoch najneskôr možno dosiahnuť úrodu obsahujúcu presne 4247 plodov?

Citrónová sada

Citrónová sada obsahuje ešte dyňu a egreš. V teplom roku vyprodukuje každá rastlina dva plody. V studenom roku je aktívny len citrón, dyňa s egrešom len zachovávajú rod. Na začiatku máme jeden citrón (C).

$$\begin{array}{cc} \hline C \xrightarrow{T} CE & C \xrightarrow{S} DE \\ D \xrightarrow{T} CE & D \xrightarrow{S} D \\ E \xrightarrow{T} EE & E \xrightarrow{S} E \\ \hline \end{array}$$

Ak by sme chceli dosiahnuť 9 plodov, mohli by sme to spraviť napríklad takto:

$$C \xrightarrow{S} DE \xrightarrow{T} CEEE \xrightarrow{T} CEEEEEEE \xrightarrow{S} DEEEEEEE$$

- c) (1 bod) Nájdite takú postupnosť teplých a studených rokov, že úroda po posledom roku bude obsahovať presne 42 plodov.
- d) (2 body) Nájdite takú postupnosť teplých a studených rokov, že úroda po posledom roku bude obsahovať presne 4247 plodov.
- e) (4 body) Vieme zostrojiť vhodnú postupnosť rokov pre úrodu každej veľkosti? Pokiaľ áno, vysvetlite, ako by ste takúto postupnosť zostrojili. Pokiaľ nie, uveďte protipríklad a vysvetlite, prečo sa nedá dopestovať dané množstvo plodov.

Fibonacciho sada

Fibonacciho sada počas teplých rokov produkuje Fibonacciho postupnosť spomenutú v minulom kole. Na studený rok je však extrémne citlivá. Začíname s rastlinou F a pravidlá budú totožné s tými použitými ako príklad na začiatku tejto úlohy.

$$\begin{array}{cc} \hline F \xrightarrow{T} G & F \xrightarrow{S} 0 \\ G \xrightarrow{T} FG & G \xrightarrow{S} G \\ \hline \end{array}$$

Ak by sme chceli dosiahnuť 9 plodov, mohli by sme to spraviť napríklad takto:

$$F \xrightarrow{T} G \xrightarrow{T} FG \xrightarrow{T} GFG \xrightarrow{T} FGGFG \xrightarrow{S} GGG \xrightarrow{T} FGFGFG \xrightarrow{T} GFGGFGGFG$$

- f) (2 body) Nájdite takú postupnosť teplých a studených rokov, že úroda po posledom roku bude obsahovať presne 42 plodov.
- g) (3 body) Vieme zostrojiť vhodnú postupnosť rokov pre úrodu každej veľkosti? Pokiaľ áno, vysvetlite, ako by ste takúto postupnosť zostrojili. Pokiaľ nie, uveďte protipríklad a vysvetlite, prečo sa nedá dosiahnuť.

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 netriviálna úloha.

3. Pofidérnejšie kasíno

15 bodov za riešenie

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

Po tom, ako ste odhalili chyby v kasíne z minulej série a takmer ste nás dovedli do krachu, naši bezpečnostní experti spravili v kasíne mnoho zmien. Už nestačí uhádnuť správny tip len raz, ale z daného počtu pokusov musíte uhádnuť aspoň nejaký minimálny počet (pre každú hru iný). Zároveň vám už nehovoríme, aké čísla naše kasíno vygenerovalo a ani neprezradíme, ako presne je kasíno implementované.

Ďalšie bezpečnostné opatrenia zahŕňajú to, že do kasína už nemôžete vstupovať osobne, ale posielate tam svojho robota, ktorý bude hrať za vás.

Úloha

V kasíne sú tri hry označené A, B, C. Princíp fungovania týchto hier je rozpísaný na konci tohto zadania. Vaším cieľom je vymyslieť a naprogramovať stratégiu, ktorá bude schopná správne uhádnuť dosť veľa tipov v danej hre.

Jednotlivé hry môžete hrať cez netcat, podobne ako v predošlej podúlohe, avšak teraz **za hranie cez netcat nezískavate body**. Služi to len na to, aby ste si mohli vyskúšať, ako jednotlivé hry fungujú. Návod na inštaláciu programu netcat nájdete [v zadaní predošlej série](https://prask.ksp.sk/ulohy/zadania/1344/)³. Spustíte ho nasledovnými príkazmi:

```
-- Linux / Mac OS
nc 158.195.16.154 9948
-- Windows
cd cesta_k_netcatu
nc.exe 158.195.16.154 9948
```

³<https://prask.ksp.sk/ulohy/zadania/1344/>

Akonáhle sa s hrami zoznámite a rozmyslíte si, ako budete hry hrať, môžete vašu stratégiu naprogramovať. Tu si môžete stiahnuť ukázkové programy C++⁴ alebo v Pythone⁵, ktoré sme pre vás pripravili, aby ste sa nemuseli trápiť komunikáciou s kasínom. Použitie týchto šablón je dobrovoľné, môžete si kľudne vytvoriť kompletne vlastné programy. Ak by ste chceli, aby sme vám vytvorili ukázkové programy aj v iných programovacích jazykoch, **napište Prefixovi**.

Na odovzdanie programu nemusíte vyriešiť všetky podúlohy, ani ich nemusíte riešiť v poradí. Kľudne odovzdajte aj program, ktorý rieši len časti niektorých podúloh. Za každú hru môže program získať 3 body, jeden za každý level. Level je len skupina vstupov, pre ktoré platí nejaké obmedzenie (napríklad, že čísla na vstupe sú menšie ako 5).

Dôležité upozornenie: do výsledného bodovania sa počíta len posledný odovzdaný program. To znamená, že ak aj odovzdáte program, ktorý úspešne rieši hru A, ale ako posledný odovzdáte program, ktorý rieši iba hru B, tak sa vám zarátajú body len za hru B. Preto sa uistite, že ako posledný pred koncom série ste odovzdali program, ktorý rieši všetky podúlohy, ktoré chcete odovzdať. Program môžete odovzdať najviac stokrát, ak by ste chceli odovzdať viackrát, najprv sa ozvite Prefixovi. Pre úplnosť doplníme, že beh vášho programu je obmedzený na 2 sekundy, čo by vás ale nemalo výrazne obmedzovať, lebo v skutočnosti stačí oveľa kratší čas.

Okrem programov odovzdajte aj popis riešenia. Popisy všetkých hier, ktoré ste riešili odovzdajte spoločne v jednom súbore, pretože opravovatelia uvidia len posledný odovzdaný súbor. Do popisu stručne napíšte, **ako ste jednotlivé hry riešili**. Aké úvahy a výpočty vás dovedli ku tej stratégii, ktorú ste nakoniec naprogramovali? Prečo váš program tipoval práve tie hodnoty aké tipoval?

V hrách A (kocky) pre $n = 6$ a B (dvere) pre $n = 5$ a $k = 2$ navyše vypočítajte, **aká je pravdepodobnosť** uhádnutia správnej hodnoty v jednom kole (t.j. keby ste hrali veľmi veľa tipov, akú časť z nich by ste priemerne uhádli)? Ak túto pravdepodobnosť správne spočítate aj v hre C (mince) pre $n = 7$, $c_1 = 1$, $c_2 = 2$, $m_1 = 3$, $m_2 = 4$, môžete získať jeden bonusový bod.

Pravdepodobnosť, nemusíte vypočítať ručne a môžete na jej výpočet použiť program. V takom prípade skopírujte zdrojový kód použitého programu do popisu riešenia. Nestačí napísať výsledok, chceme vidieť aj postup, výpočet alebo program, ktorý vás k výsledku dovedol. V každej hre môžete za popis získať najviac 2 body.

Ak budete mať akékoľvek **otázky ohľadom úlohy**, toho ako funguje netcat, Python, C++, samotné odovzdávanie alebo budete chcieť spraviť viac ako sto submitov, **napište Prefixovi**. Rád vám pomôže.

Hra A – kocky (5 bodov)

V kasíne máme štyri n -steny. Každý n -sten má na stranách čísla od 1 po n . Ak má napríklad 7 stien, tak má na stranách čísla od 1 po 7. Môžete si to predstaviť ako hracie kocky s n stranami. T -krát hodíme týmito štyrmi n stenami a vy si máte po každom hode tipnúť súčet čísel, ktoré na týchto n stenoch padli. Následne vyhodnotíme, ako dobre sa vám darilo a podľa toho vám pridáme body.

Upresnenia: Na prvom riadku vstupu je písmeno A, na druhom číslo n (počet stien) a na treťom T (počet tipov). V prvom leveli platí, že $n = 6$. V druhom leveli sme boli zhovievaví, a netreba uhádnuť až tak veľa tipov. Počet stien však môže byť hocikolko od 6 do 10. V treťom leveli musíte hádať najlepšie ako viete a n leží medzi 6 a 10.

Príklad:

vstup	výstup
A	1
6	1
5	1
	1
	1
	1

Poznámka, takýto výstup by dostal 0 bodov, pretože na 4 kockách padne vždy súčet aspoň 4. Ak tipnete 1, neuhádnete.

Hra B – dvere (5 bodov)

V kasíne máme n dverí, za práve jednými je schovaná výhra. Najprv vy ukážete na jedny dvere a potom my spomedzi zvyšných dverí otvoríme k , za ktorými nie je schovaná výhra. Potom je vašou úlohou tipnúť dvere s výhrou (môžete tipovať aj tie, na ktoré ste ukázali ako prvé). Takto si zahráme T hier, pričom na začiatku každej

⁴<http://media.ksp.sk/ulohy/prask/3/4/3/sablona.cpp>

⁵<http://media.ksp.sk/ulohy/prask/3/4/3/sablona.py>

hry náhodne vyberieme dvere s výhrou (každé dvere majú na začiatku rovnakú šancu byť výherné) a budeme počítať, koľkokrát sa vám ich podarí uhádnuť.

Upresnenia: Na prvom riadku vstupu je písmeno B, na druhom sú čísla n a k a na treťom riadku je číslo T (počet hier). Následne odohráme postupne T hier, v každej najskôr váš program vypíše číslo (číslo dverí), my vám napíšeme zoznam k dverí, ktoré sú bez výhry a váš program tipne, za ktorými dverami je výhra.

V prvom leveli platí, že $n = 3$, $k = 1$, v druhom leveli $3 \leq n \leq 8$, $k = 1$. V treťom leveli $3 \leq n \leq 10$, $1 \leq k \leq n - 2$.

Príklad:

vstup	výstup
B	5
5 2	2
1	
1 2	

Tentokrát mala hra len jedno kolo, program ukázal na piate dvere, dozvedel sa, že v dverách 1 a 2 sa nič nenachádza. Napriek tomu tipol druhé dvere a teda opäť nevyhral.

Hra C – mince (5 bodov + bonus)

V kasíne máme n mincí a každá má na jednej strane číslo c_1 a na druhej c_2 . Zároveň si vyberieme nejaké čísla m_1 a m_2 . Hodíme všetkými mincami a sčítame čísla na vrchných stranách (mince sú férové a nepadajú na hranu). Prezradíme vám aký je zvyšok súčtu po delení m_1 a spýtame sa vás na zvyšok súčtu po delení m_2 . Inak povedané, ak si označíme súčet s , tak vám prezradíme $s \% m_1$ a vy musíte uhádnuť $s \% m_2$.

Upresnenia: Na prvom riadku vstupu je písmeno C, na druhom sú čísla n, c_1, c_2, m_1, m_2 a na treťom riadku je číslo T (počet hier). Následne odohráme postupne T hier, v každej vám povieme prvý zvyšok a vy tipnete druhý zvyšok.

V prvom leveli platí, že $n = 1$, v druhom leveli $n = 5$ a zároveň sme menej prísni v tom, koľkokrát musíte správne tipnúť odpoveď. V treťom leveli $1 \leq n \leq 10$, a treba uhádnuť čo najviac tipov. Vo všetkých troch leveloch platí, že $1 \leq c_1, c_2, m_1, m_2 \leq 10$

Príklad:

vstup	výstup
C	1
2 3 1 4 5	4
2	
2	
0	

Hodíme dvoma mincami, ktoré majú na stranách čísla 1 a 3. V prvom pokuse sa program dozvedá, že zvyšok súčtu po delení 4 je 2. To znamená, že súčet je 2 alebo 6, nevie však, čo z toho. Keďže su obe rovnako pravdepodobné, tipne si, že súčet bol 6 a teda vypíše $6 \% 5 = 1$. V druhom prípade má šťastie, zvyšok po delení 4 bol 0, takže program vie, že súčet bol 4. Vypíše teda $4 \% 5 = 4$

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 našim testovacím systémom, odporúčam ti zájsť na Programátorskú Liaheň (<http://betaliahen.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 nebudované 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 pribudnú na Liahni ďalšie dve sady, ktorými si budeš môcť opäť nahradiť programátorské úlohy.

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

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

4. Párty Žiab

0 bodov za riešenie

Táto úloha sa dá nahradiť riešením sady **loops_cpp** na Liahni (betaliahen.ksp.sk⁶). Ak chceš, aby ti namiesto bodov za riešenie tejto úlohy boli započítané body získané riešením spomínanej sady, na stránke odovdžaj pdf-ko s prezývkou, ktorú používaš na Liahni.

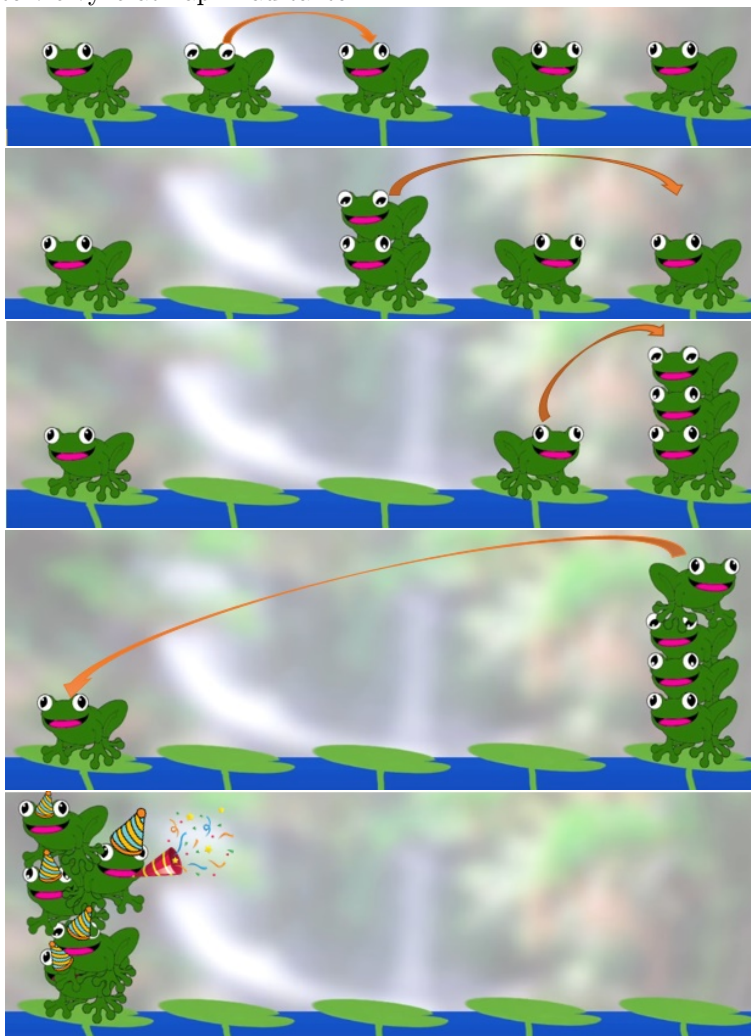
Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Paulínke na psmolarova@gmail.com

Na jazere žijú žaby. Presnejšie, každá žaba žije na svojom lekne. Lekien je na jazere presne n a na vode plávajú v rade za sebou, očíslované od 1 po n . Každý večer sa v rybníku koná párty žiab. Zorganizovať takúto párty je jednoduché – stačí, keď sa všetky stretnú na tom istom lekne.

Žaby sa však cestou na párty nechcú zamочиť, preto sa na ňu musia dostať iba pomocou skákania po leknách. No a tu to začína byť komplikované. Žiadna žaba nechce byť asociálna a preto musí vždy preskočiť na lekno, na ktorom sedí aspoň jedna ďalšia žaba. A keďže je lekno malé, žaby sa pri skákaní stavajú na seba. Ďalším skákaním sa ich poradie však nemení.

Naviac, keď je na lekne viacero žiab, skáču všetky spoločne. A z neznámych príčin, ak je na lekne x žiab, tak vedú skočiť iba na lekno, ktoré je vzdialené **presne** o x lekien doprava alebo doľava. Na začiatku, keď je žaba na svojom lekne sama, môže skočiť iba na susedné lekno.

Cesta na párty preto vie vyzeráť napríklad takto:



⁶<http://betaliahen.ksp.sk>

Ako si iste viete predstaviť, keď na chrbte niektorej žaby stojí viacero iných žiab, je to pre ňu namáhavé. A námahu veru nemá žaba Michal v láske. Preto by chcel navrhnúť takú postupnosť skokov, aby sa všetky žaby stretli na jednom lekne a on sa ocitol na vrchu kopy. Teda v žiadnom momente nemôže iná žaba (alebo viacero žiab) skočiť na lekno, na ktorom sa nachádza Michal.

Ak by teda v situácii, ktorá je znázornená na obrázku, sedel žaba Michal na štvrtom lekne, tak by mu takáto postupnosť skokov vyhovovala, keďže by nikdy na ňom nestála žiadna iná žaba. Ak by však začínal na druhom lekne, tak v treťom skoku by naňho skočila žaba z lekna štyri, čo by sa mu nepáčilo.

Pomôžte mu nájsť takúto postupnosť skokov!

Úloha

Na vstupe dostaneme počet lekiek v rybníku a pozíciu žaby Michala. Nájdite takú postupnosť skokov, že dodržiava všetky vyššie uvedené pravidlá skákania žiab a zároveň platí, že žiadna žaba neskočí na lekno, na ktorom sa nachádza žaba Michal.

Až 9 bodov môžete získať za vyriešenie ľahšej podúlohy, v ktorej sa v rybníku žaba Michal nenachádza. V tom prípade hľadáte iba takú postupnosť skokov, ktorá vedie k tomu, že všetky žaby skončia na tom istom lekne.

Vstup

Na vstupe dostanete dve čísla oddelené medzerou n a m určujúce počet lekiek a pozíciu žaby Michala.

V niektorých vstupoch bude $m = -1$. V týchto vstupoch sa žaba Michal nenachádza a teda hľadáme ľubovoľnú postupnosť skokov, po ktorej sa všetky žaby stretnú na tom istom lekne.

Výstup

Ak neexistuje žiadna platná postupnosť ktorá spĺňa podmienky popísané v úlohe, vypíšte „NIE“. V opačnom prípade vypíšte „ANO“ a následne vypíšte platnú postupnosť skokov. Každý skok je popísaný dvoma číslami a a b ($1 \leq a, b \leq n$) – všetky žaby z lekna a skočili na lekno b .

V prípade, že existuje viacero správnych postupností, ktoré vedú k párti žiab, môžete vypísať ľubovoľnú z nich.

Hodnotenie

Váš program bude spustený na piatich sadách vstupných súborov. Body dostanete za každú úspešne vyriešenú sadu. Obmedzenia na veľkosť čísla n v jednotlivých sádach nájdete v nasledujúcej tabuľke.

Číslo sady	1	2	3	4	5
maximálne n	20	1 000	100 000	1 000	100 000
obmedzovania m	$m = -1$	$m = -1$	$m = -1$	bez obmedzenia	bez obmedzenia

Príklady

vstup

5 -1

výstup

ANO
2 3
3 5
4 5
5 1

Tento vstup zodpovedá obrázku v zadaní.

vstup

9 3

výstup

ANO
1 2
3 2
4 5
6 5
7 8
9 8
8 5
2 5

5. Poriadnejšia výzva

15 bodov za riešenie

Táto úloha sa dá nahradiť riešením sady `arrays_cpp` na Liahni (betaliahen.ksp.sk⁷). Ak chceš, aby ti namiesto bodov za riešenie tejto úlohy boli započítané body získané riešením spomínanej sady, na stránke odovdaj pdf-ko s prezývkou, ktorú používaš na Liahni.

Ak máte akékoľvek otázky ohľadom tejto úlohy napíšte Andrejovi na andrej.korman1@gmail.com

Výborne! Ani my nevieme skryť nadšenie z toho, koľkí z vás si poradili s našou výzvou – vyriešiť osemsmernovky a pomôcť tým Žabovej babke. Dokonca ste si tým vyslúžili aj jej rešpekt. Nastal však problém, ako to už býva, babka o osemsmernovky z tohto dôvodu stratila záujem a teraz nevie, čo robiť. Nádej prišla až včera ráno, keď zahliadla vo svojich obľúbených novinách nový typ hlavolamu s názvom sudoku. Zo začiatku si s ním nevedela dať rady, ale po chvíli v ňom už bola expertka.

Rozhodla sa, že si s vami tentokrát dá súťaž, keď už vám tie hlavolamy tak idú. Prebiehať to bude tak ako ste zvyknutí. Vy odovzdáte program a ona začne v tom istom čase riešiť. Ak to stihnete skôr ako ona, body sú vaše, ak nie, tak za daný pokus nič nedostanete. Babke trvá vyriešenie každej sudoku presne 2 sekundy. Vďaka súbojom s vami možno príde na lepšie metódy riešenia a tie vám po konci tejto série pošle, ako námet na riešenie.

Úloha

Sudoku⁸ je hra, v ktorej dostanete mriežku rozmerov 9×9 , v ktorej sú zo začiatku doplnené nejaké čísla a nejaké samozrejme chýbajú. Vašou úlohou je doplniť všetky prázdne políčka podľa nasledujúcich pravidiel:

1: do mriežky môžete doplniť iba čísla 1 až 9

2: pre každý riadok musí platiť, že sa v ňom nachádza každé číslo práve raz

3: pre každý stĺpec musí platiť, že sa v ňom nachádza každé číslo práve raz

4: ak si rozdelíme túto mriežku 9 štvorcov veľkosti 3×3 tak ako na obrázku, musí platiť, že v každom sa nachádza každé číslo práve raz.

Čísla, ktoré sú na začiatku napísané vo vnútri mriežky samozrejme spĺňajú hore uvedené pravidlá.

Sudoku môže vyzerať napríklad nasledovne.

	2	8	1	6				4
			4	7				
					8	9		1
4		5					1	
				1				
	3					4		6
6		3	9					
				8	7			
9				4	5	8	3	

Na vstupe od nás dostanete jedno takéto sudoku s niekoľkými prázdnyimi políčkami. Vašou úlohou je napísať program, ktorý dané sudoku vyrieši a vypíše na výstup.

Vstup

Na vstupe sa bude nachádzať presne 9 riadkov, pričom každý bude obsahovať 9 čísel. j -te číslo na i -tom riadku $a_{i,j}$, reprezentuje číslo v i -tom riadku a j -tom stĺpci mriežky. Pre všetky $a_{i,j}$ platí $0 \leq a_{i,j} \leq 9$. Iste ste si

⁷<http://betaliahen.ksp.sk>

⁸<https://sk.wikipedia.org/wiki/Sudoku>

všimli, že $a_{i,j}$ je z rozsahu až 10 čísiel. Prázdne políčko budú reprezentované číslom 0. To znamená, že doplniť budete rovnako ako v pôvodnom sudoku iba čísla 1 až 9.

Výstup

Na výstup vypíšete správne doplnenú sudoku. Vašou úlohou je teda nahradiť všetky 0 zo vstupu tak, aby boli dodržané hore uvedené pravidlá. V prípade, že dané sudoku má viac riešení, vypíšete ľubovoľné z nich. Môžete predpokladať, že každé zadané sudoku má aspoň jedno riešenie.

Sudoku vypisujete na výstup po riadkoch. Medzi jednotlivými číslami sa musia nachádzať medzery. To, či sa aj za posledným číslom v riadku bude nachádzať medzera alebo nie, je na vás, nespôsobí to chybu pri testovaní, avšak nezabudnite každý riadok oddeliť znakom konca riadku “\n”. Výstup má teda obsahovať 9 riadkov, pričom v každom sa bude nachádzať presne 9 čísel oddelených medzerou, reprezentujúcich jednotlivé políčka v sudoku, za ktorými bude nasledovať znak konca riadku.

Hodnotenie

Vaše riešenie bude ako obvyčajne testované na rôznych sádach, ktoré sa budú líšiť obtiažnosťou ich riešenia. Keďže riešiť menšie sudoku by bolo podvádzanie, sady sa líšia iba tým, koľko čísel v sudoku chýba.

- V prvej sade platí, že v každom riadku chýba práve jedno políčko.
- V druhej sade platí, že v každom stĺpci chýba práve jedno políčko.
- V tretej sade platí, že v každom bloku veľkosti 3x3 chýba práve jedno políčko.
- Vo štvrtej sade platí, že počas riešenia a dopĺňania vždy existuje aspoň jedno políčko, ktoré vieme jednoznačne doplniť z pohľadu na riadok, stĺpec a blok, v ktorom sa nachádza. Inak povedané, pre toto políčko iba jedno číslo spĺňa, že sa nenachádza v rovnakom stĺpci, riadku a bloku.
- V piatej sade neplatia žiadne obmedzenia.

Príklady

vstup

```
0 2 8 1 6 0 0 0 4
0 0 0 4 7 0 0 0 0
0 0 0 0 0 8 9 0 1
4 0 5 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0
0 3 0 0 0 0 4 0 6
6 0 3 9 0 0 0 0 0
0 0 0 0 8 7 0 0 0
9 0 0 0 4 5 8 3 0
```

výstup

```
7 2 8 1 6 9 3 5 4
5 1 9 4 7 3 6 2 8
3 4 6 2 5 8 9 7 1
4 9 5 8 3 6 2 1 7
8 6 2 7 1 4 5 9 3
1 3 7 5 9 2 4 8 6
6 8 3 9 2 1 7 4 5
2 5 4 3 8 7 1 6 9
9 7 1 6 4 5 8 3 2
```

Vstup a výstup zodpovedá osemsmierovke zo zadania spolu s riešením.

vstup

```
0 7 2 3 9 5 6 4 1
4 0 6 7 2 1 8 9 5
9 5 0 6 8 4 2 3 7
7 1 3 0 4 2 9 5 6
2 8 5 9 0 6 1 7 4
6 4 9 1 5 0 3 2 8
5 2 8 4 6 3 0 1 9
1 6 4 2 7 9 5 0 3
3 9 7 5 1 8 4 6 0
```

výstup

```
8 7 2 3 9 5 6 4 1
4 3 6 7 2 1 8 9 5
9 5 1 6 8 4 2 3 7
7 1 3 8 4 2 9 5 6
2 8 5 9 3 6 1 7 4
6 4 9 1 5 7 3 2 8
5 2 8 4 6 3 7 1 9
1 6 4 2 7 9 5 8 3
3 9 7 5 1 8 4 6 2
```

Vstup, ktorý sa môže vyskytnúť aj v prvej aj v druhej sade.