

Zadanie školského kola súťaže ZENIT v programovaní

Kategória A a B, 17.10.2023

V prípade nejasností konzultujte záložku **Pomoc** na stránke zenit.ksp.sk, alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

A: Alčka má pre vás úlohu

10 bodov

Túto úlohu pripravil nemenovaný Large Language Model

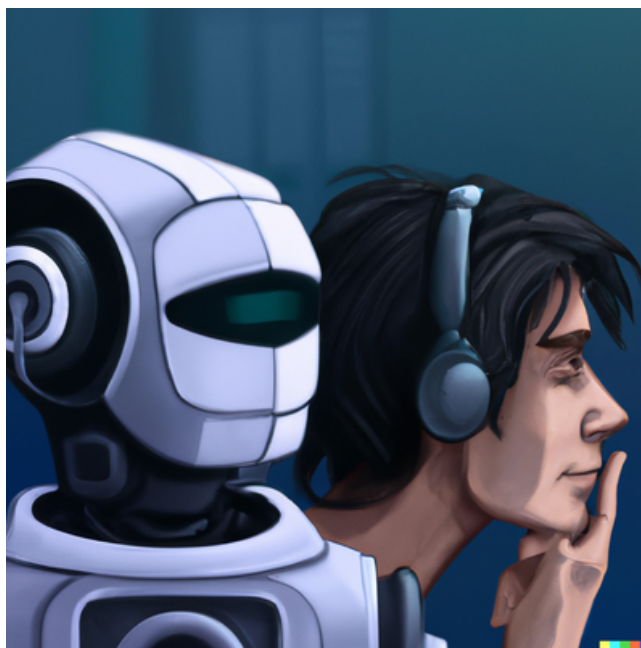
Raz dávno v krajine kódu, v meste Algoria, žil šikovný programátor menom Alan. Alan bol obľúbený v celej Algórii pre svoje zručnosti a múdrosť v kóde. Jedného dňa, keď Alana navštívil tajomný robot s menom Robi, zistil o záhadnej úlohe, ktorá ho čakala.

Robi šepkal Alanovi do ucha: “V tvojom kóde sa skrýva tajomstvo, najdlhšie slovo ti odhalí cestu. Ale choď opatrne, môže to byť ťažký úkol.”

Alan sa zatváril vážne a bez váhania sa pustil do práce. Začal prechádzať kód riadok po riadku, hľadajúc tú správnu cestu. Každý riadok bol plný rôznych premenných, cyklov a podmienok. Alan sa nemohol nechať zmiast, musel nájsť to najdlhšie slovo, čo i len jediné.

Po niekoľkých hodinách pátrania Alan konečne narazil na slovo, ktoré malo nezameniteľný lesk. “Európska”, zaznelo ticho z jeho úst, keď napokon objavil to najdlhšie slovo, ktorému sa venoval celý jeho úsilný prieskum.

A tak, s najdlhším slovom v jeho kóde, Alan odhalil tajomstvo, ktoré mu odhalilo nový svet možností a algoritmov. Robi s úctou naklonil hlavu a odviedol Alana do nového sveta, kde boli len možnosti a žiadne hranice. A Alan zostal zapamätaný ako hrdina, ktorý našiel to najdlhšie slovo a otvoril cestu novému veku programovania v Algórii.



Úloha

Napíšte program, ktorý identifikuje a vypíše najdlhšie slovo v preddefinovanej rozprávke.

Samotná úloha spočíva v ručnom skenovaní a čítaní rozprávky s cieľom identifikovať najdlhšie slovo v texte. Proces vyžaduje ručné analyzovanie každej vety a identifikáciu slov s najväčším počtom písmen. V závislosti od dĺžky rozprávky a počtu slov môže táto úloha vyžadovať pozornosť a trpezlivosť.

Pri vykonávaní tejto úlohy by ste mali postupovať nasledovne:

1. Prečítajte si rozprávku pozorne a postupne analyzujte každé slovo.
2. Pri výskyte slova s väčším počtom písmen si zaznamenajte toto slovo.
3. Pokračujte v čítaní rozprávky a aktualizujte záznam o najdlhšom slove, ak nájdete slovo s ešte väčším počtom písmen.
4. Keď dokončíte čítanie celej rozprávky, vypíšte zaznamenané najdlhšie slovo.

Všetky diakritické znaky (čiže háčiky, čiarky, bodky nad písmenami atď.) majú byť ignorované, a preto sa má text pretransformovať do formy bez diakritiky. Ak sa vyskytne viacero slov s rovnakým maximálnym počtom písmen, program môže vypísať ľubovoľné z týchto slov.

Vstup a Výstup

Program nevyžaduje žiadny vstup. Funguje automaticky bez potreby interakcie s používateľom.

Príklad

vstup	výstup
<input type="text"/>	<input type="text" value="Europska"/>

Toto možno bolo najdlhšie slovo v Alanovom kóde, nie je to však najdlhšie slovo v našej rozprávke, ktoré hľadáte.

B: Bardejovská zmrzlina

15 bodov

Kamaráti Miro a Patrik z Bardejova išli po školskom kole Zenitu spolu na obed. Po ceste Miro uvideli zmrzlináreň, tak sa spýtal Patrika či nenenene-chce zmrzlinu? Patrik zmrzlinu chce, ale nevie, čo mu má odpovedať aby to aj Miro pochopil.

Úloha

Na prvom riadku je otázka.

Odpovedzte tak, aby Miro pochopil že chcete zmrzlinu.

Vstup a Výstup

Na prvom riadku je otázka vo forme [ne]-chces zmrzlinu?, pričom [ne] znázorňuje aspoň jedno ale najviac 100 000 ne.

Odpovedzte na ňu ano alebo nie tak, aby ste vyjadrili záujem o zmrzlinu.

Príklady

vstup	výstup
<input type="text" value="ne-chces zmrzlinu?"/>	<input type="text" value="nie"/>

Patrik chce zmrzlinu, čiže nie ne-chece zmrzlinu.

vstup	výstup
<input type="text" value="nene-chces zmrzlinu?"/>	<input type="text" value="ano"/>

Patrik chce zmrzlinu, čiže ano nene-chece zmrzlinu.

C: Chcem mať veľa kamarátov

20 bodov

Miloš má strašne veľa kamarátov. Každý rok míňa svoje ťažkozarobené peniaze, aby každému svojmu kamarátovi na narodeniny kúpil darček. Priateľstvá sú však vrtkavá záležitosť, Miloš nikdy nevie, či sa s ním niekto neprestane kamarátiť kvôli tomu, že je príliš múdry.

Miloš chce vrtkavosť svojich kamarátstiev zohľadniť, keď si bude robiť finančnú uzávierku roku 2023. Má zoznam, v ktorom má pre každého kamaráta sumu, ktorú minul na jeho darček.

Miloš by chcel vedieť, koľko peňazí by musel minúť na darčeky v prípade, že by sa s ním ktorýkoľvek z jeho kamarátov prestal kamarátiť, lebo je príliš múdry. No nemá čas tento údaj vypočítať pre každého svojho kamaráta, lebo je príliš zaneprázdnený hľadaním si nových kamarátov.

Skúste mu s tým pomôcť, možno sa s vami bude kamarátiť. Teda, ak sa nebudete báť jeho múdrosti.

Úloha

Na vstupe budete mať v jednom riadku počet Milošových kamarátov. V ďalšom riadku je zoznam, v ktorom má pre každého z týchto kamarátov číslo - sumu, ktorú tento rok minul na darček daného kamaráta. Pre každého z jeho kamarátov napíšte, koľko by stáli darčeky všetkých ostatných kamarátov okrem neho.

Vstup a Výstup

Na prvom riadku je celé číslo n reprezentujúce počet Milošových kamarátov. V ďalšom riadku sa nachádza zoznam n kladných čísel c_1 až c_n , neprevyšujúcich 1000: ceny darčiekov každého kamaráta.

Pre prvú sadu platí, že $1 \leq n \leq 10^3$. Pre druhú sadu $1 \leq n \leq 2 \cdot 10^5$.

Pre každého kamaráta vypíšte na samostatný riadok jedno číslo: koľko by stáli dokopy všetky darčeky, keby sme nekúpili darček práve jemu.

Príklad

vstup	výstup
5 1 2 3 4 6	15 14 13 12 10

Ak by sa Miloš prestal kamarátiť s prvým kamarátom, na darčeky by minul 15 peňazí. Ak by sa prestal kamarátiť s druhým kamarátom, minul by 14 peňazí. Rovnaký princíp platí pre všetkých kamarátov.

D: Deadline

25 bodov

Bolo to presne pred mesiacom keď som dostal za úlohu vybrať tých nesmierne dôležitých N čísel. A bolo to včera večer, keď som si uvedomil, že som konečne našiel tie správne čísla. Tie osudové čísla, ktoré zabezpečia úspech našej firme. Dôkladne som ich vybral na základe analýz a štatistík, a cítil som sa, akoby som práve objavil tajomstvo života. Dokonca akurát načas, keďže dnes ich musím odovzdať nadriadenému Barboriakovi.

Už ako som išiel výsledky svojej mesačnej práce poslať, doslova pár minút pred tým ako sa mala táto epizóda môjho života skončiť, mi prišla správa. "Ahoj, potrebujeme úplne opačné čísla! Verím a očakávam, že výsledky obdržím v dohodnutom termíne. Barboriak."

Ten termín je dnes o jednej a oni chcú úplne opačné čísla? Do pekla aj s takouto spoluprácou!

Úloha

Vašou úlohou je z N čísel spraviť opačných N čísel. Opačné číslo má poradie bitov obrátené. Všetky čísla sú 32-bitové bez znamienka.

Vstup a Výstup

Na prvom riadku vstupu bude číslo n ($1 \leq n \leq 100$) udávajúce počet čísel na vstupe. Nasleduje n riadkov, každý obsahuje jedno číslo x v desiatkovej sústave ($0 \leq x < 2^{32}$).

Príklad

vstup	výstup
3 1 5 8	2147483648 2684354560 268435456

Binárny zápis 32-bitového čísla 1 je 00000000000000000000000000000001 a jeho opačné číslo je 10000000000000000000000000000000 čo je 2147483648 v desiatkovej sústave.

E: Extrémna prokrastinácia

30 bodov

Miško by mal pripravovať úlohu do Zenitu. Viete ale ako to chodí, nechce sa mu a tak robí všetko iné, len nie pripravovať úlohu. Našiel si n filmov, ktoré by si chcel pozrieť. Nie je to však také jednoduché. Miško je pokazený dnešnou dobou, a sociálnymi sieťami, takže neudrží pozornosť len na jeden film. Našťastie má 2 obrazovky, a teda môže pozerať dva filmy naraz.

To však stále nestačí. Je dokázané že každých 5 minút je v každom filme nudná scéna. Na to však Miško má tiež riešenie. Keď začne nudná scéna (pochopteľne naraz v oboch filmoch ktoré pozerá) pozrie si trailer nejakého iného filmu.

Úloha

Miška by teraz zaujímalo ako najdlhšie dokáže udržať pozornosť na filmy, a tým oddialiť jeho prácu na úlohách Zenitu. Miško si najprv vyberie 2 filmy ktoré bude pozeráť, a potom každých 5 minút si pozrie trailer na nejaký film, ktorý zatiaľ nevidel, ani ho nepozera. Akonáhle jeden z filmov skončí, alebo začne nudná scéna a Miško už nemá čo pozeráť, začne sa nudiť.

Vstup a Výstup

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10^5$) udávajúce počet filmov. Ďalej nasleduje n riadkov – každý z nich obsahuje jedno celé číslo a_i , ktoré zodpovedá dĺžke i -teho filmu v 5-minútovkách, pričom platí $1 \leq a_i \leq 10^6$.

Vypíš jeden riadok a v ňom jedno celé číslo k - najväčší počet trailerov, ktoré si Miško pozrie, kým sa začne nudiť.

Príklad

vstup	výstup
5 6 1 4 8 2	3

Miško začne pozeráť filmy s dĺžkou 6 a 8. Potom si postupne pozrie trailery na zvyšné tri filmy.

vstup	výstup
4 2 1 2 1	1

Ak Miško začne pozeráť filmy s dĺžkou 2 stihne si pozrieť len jeden trailer. Inak by si nestihol pozrieť ani jeden, kým sa skončí film, ktorý pozerá.

F: Fuška

40 bodov

Z nejakého divného dôvodu doručí každé ráno biela sova organizátorom zenitu v programovaní osemsmerevku.

Žiaden z nich sa nepriznal k zapísaniu na sovi osemsmerevkový mailinglist. Pokusy o komunikáciu so sovou v snahe zrušiť odber zatiaľ zlyhali, a tak ich už majú peknú kôpku.

Aby mali aspoň nejaké využitie, rozhodli sa, že namiesto hodenia si mincou pri rozhodovaní, ktorým submitom dať OK, sa pozrú, či v nejakej osemsmerevke vedia nájsť slovo ZENIT.

Osemsmerevky však vedia byť zákerné.

Viete napríklad v nasledovnej osemsmerevke nájsť psa po anglicky, teda DOG?

```
DG00DD0DGO0DD0
0D00GGD0D0G0GG
0G0GD00DGO0DDD
DG000GG00GDGO
0GDG0GDG0GG0GD
DDDGDD0D00GD00
0DG0GGD00GG00D
```

Veru, riadna fuška.

O fuškách platí, že sú na ne organizátori zenitu v programovaní akútne alergickí.

Preto ZENIT v osemsmerevkách rozhodne hľadať nebudú, budete to robiť vy.

Úloha

Daná je osemsmierovka, teda matica písmen anglickej abecedy.
Zistite, či sa v nej nachádza (v niektorom z ôsmich smerov) slovo ZENIT.

Vstup a Výstup

Prvý riadok obsahuje číslo n - počet riadkov a stĺpcov osemsmierovky.
Nasleduje n riadkov po n znakoch - osemsmierovka. Znaký budú veľké písmená anglickej abecedy.
V prvej sade platí $n = 5$. V druhej $1 \leq n \leq 50$.
Ak sa v osemsmierovke nachádza slovo ZENIT, vypíšte OK. Inak vypíšte WA.

Príklady

vstup

```
5
TAKTO
NIEJE
LENZE
KUPEL
MOTUZ
```

výstup

```
OK
```

Na hlavnej diagonále je obrátené ZENIT

vstup

```
6
HALOOZ
KSPJEE
SUPERN
PODTEI
RIESIT
WKSPSK
```

výstup

```
OK
```

V poslednom stĺpci

vstup

```
5
ZENTI
EIZNT
NIZET
ETNIZ
INTEZ
```

výstup

```
WA
```

G: Gardenerov gadžet

48 bodov

Život na dedine je strašná otrava. Vkuse dakomu narastú voňavejšie paradajky, rovnejší plot, výkonnejší traktor, . . . no vecí sa nájde neúrekom. Výnimkou nie je ani Adam. Jeho sused má tak krásne pole, že mu každý závidí. Aj Adam by také chcel. . . Ešte že od Klubu Sejačov a Pestovateľov na narodeniny dostal Ultimátny záhradný gadžet 3000, pomocou ktorého vie jedným šmahom zvýšiť úrodnosť ľubovoľných úsekov jeho poľa. A nie len to, dokonca ich aj ľubovoľne preusporiada!

Úloha

Na vstupe dostaneme 2 polia A a B , v každom je n čísel reprezentujúce úrodnosti daných úsekov. Vašou úlohou je zistiť, či sa dá pole A pomocou jedného použitia Adamovho gadžetu zmeniť tak, aby bolo rovnaké ako pole B . Teda aby pre všetky indexy i platilo, $a_i = b_i$. Jedno využitie Adamovho gadžetu sa skladá z 2 krokov. Najprv si vyberieme ľubovoľne veľa indexov i , na ktorých zvýšime úrodnosť úseku a_i o 1. Potom preusporiadame úseku poľa A do ľubovoľného poradia.

Vstup a Výstup

Prvý riadok vstupu obsahuje číslo n ($1 \leq n \leq 10^5$), ktoré označuje počet prvkov v poliach A a B . Nasleduje n čísel a_i ($a_i \leq 10\,000$), označujúce úrodnosť i -teho úseku Adamovho poľa. Na treťom riadku je n čísel b_i ($b_i \leq 10\,000$), označujúce úrodnosť i -teho úseku susedovho poľa.

Na jeden riadok vypíšete **Jednoduche**, ak sa dá jedným ťmahom adamovho gadžetu preorať jeho pole tak, aby bolo rovnaké ako susedove. Ak sa to nedá, vypíšete **Neda sa**.

Príklady

vstup

```
5
1 0 5 4 1
0 2 1 4 6
```

výstup

Jednoduche

Môžeme napríklad zvýšiť úrodnosť prvého a tretieho úseku. Potom len vymeníme prvý úsek s druhým a tretí s piatym.

vstup

```
5
0 5 4 2 5
2 1 6 4 4
```

výstup

Neda sa

Neexistuje spôsob, ako porať prvé pole tak, aby vyzeralo ako druhé.

H: Heslo

60 bodov

Dávid je paranoidný a bojí sa, že nepoctiví riešitelia by sa mohli snažiť získať vzorové riešenia Zenitu ešte pred súťažou. Čo ak nejaký riešiteľ ukradne mobil od jedného organizátora a nájde v ňom všetky úlohy aj s riešeniami? Toto nemôže dopustiť!

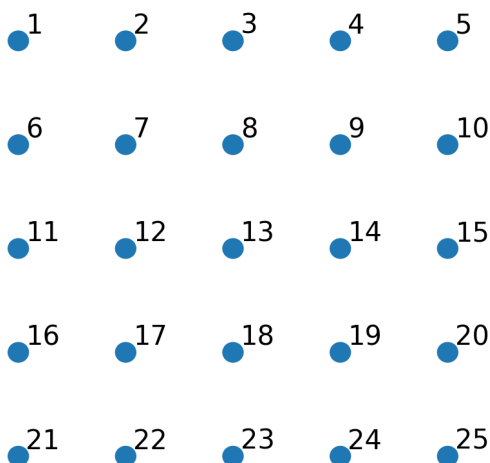
Dohodol sa s organizátormi, že si všetci nastaví na mobile odomykací vzor a aby to nemali také ťažké, rovno im nejaké vzory vymyslel. Dávid ale asi nevie ako presne mobily fungujú a preto niektoré vzory čo vymyslel nie sú platné. Potrebuje vašu pomoc (tých poctivých riešiteľov) aby ste mu pomohli určiť ktoré vzory sú platné a ktoré nie.

Úloha

Na vstupe dostanete popis vzoru a vašou úlohou je vypísať či je platný.

Základ každého vzoru je mriežka $n \times m$ bodiek (n riadkov a m stĺpcov). Vzor je nejaká postupnosť týchto bodiek spojených úsečkami, a je platný ak spĺňa nasledujúce podmienky: 1. Každá bodka je použitá najviac raz. 1. Ak úsečka medzi bodkami a a b prechádza cez inú bodku c , táto bodka sa musí v postupnosti nachádzať skôr ako a /alebo b .

Bodky budeme číslovať od 1 do $n \cdot m$ po riadkoch zhora dole a zľava doprava, tak ako na obrázku.



Vstup a Výstup

Prvý riadok obsahuje tri kladné celé čísla n, m, l popisujúce rozmery mriežky a dĺžku vzoru. Platí $1 \leq l \leq$

$n \cdot m$.

Druhý riadok obsahuje l čísel medzi 1 a $n \cdot m$ vrátane. Tie opisujú čísla bodiek z ktorých sa skladá vzor v danom poradí.

Ak je vzor platný vypíšte **Ano**, inak vypíšte **Nie**.

V prvej sade platí $n = m = 3$.

V druhej sade $n, m \leq 10$.

V tretej sade $n, m \leq 500$.

Príklady

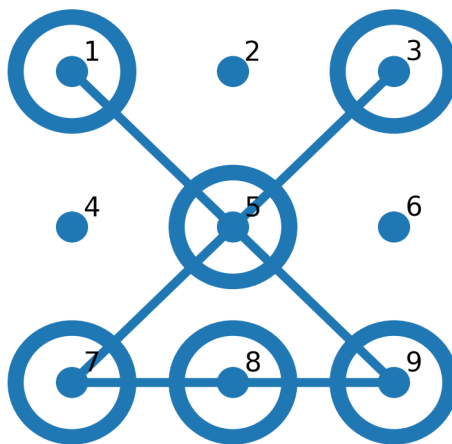
vstup

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

výstup

```
Ano
```

Vstup zodpovedá vzoru na obrázku.



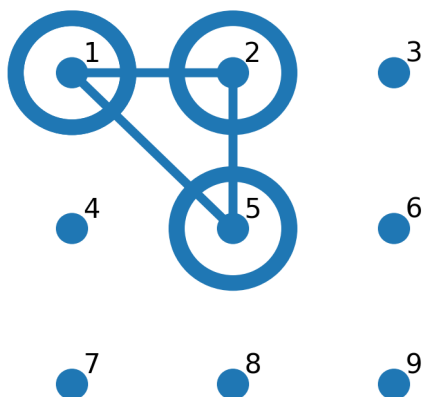
vstup

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

výstup

```
Nie
```

Bodky sa nemôžu opakovať



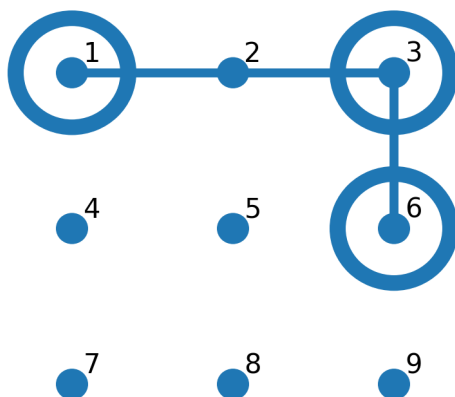
vstup

3 3 3
1 3 6

výstup

Nie

Tento vzor je tiež neplatný lebo sme prešli cez bodku číslo 2 bez toho aby sme ju označili.



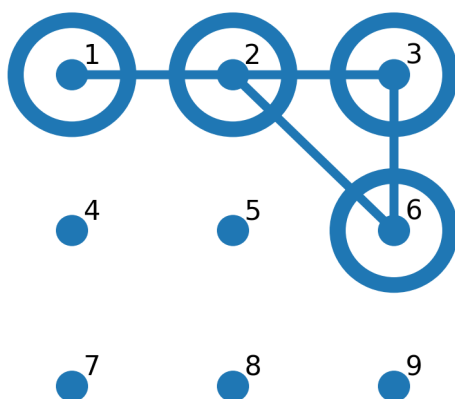
vstup

3 3 4
1 3 6 2

výstup

Nie

Aj v tomto prípade sme prešli cez bodku číslo 2 bez toho aby sme ju označili. Označili sme ju až neskôr.



I: Identifikácia identifikátoru

80 bodov

Ale však toto nie je normálne! Čo tým myslia že to neuspelo?! Je to pokazené? Som ja neschopný alebo oni hlúpi? Nakódil by som lepší softvér za dvadsať minút na školskom počítači!!!

Rozhorčujete sa po tom čo ste už hádam stí krát priložili prst na ten naničhodný čítač odtlačkov prstov. Snažíte sa pridať si prst, ale ono to proste nejde! A keď ste to už takto prehlásili, tak ten lepší softvér jednoducho budete musieť naprogramovať.

Úloha

Váš odtlačok prsta vieme reprezentovať ako mriežku, kde na každom políčku buď je alebo nie je hrbolček.

Na vstupe dostanete snímky prsta, ako ich naskenoval čítač odtlačkov prstov. Snímka je tiež reprezentovaná ako mriežka rovnakých rozmerov, kde o každom políčku vieme buď že:

1. na ňom bol hrbolček odtlačku prsta
2. na ňom nebol hrbolček odtlačku prsta
3. v tejto snímke o ňom nič nevieme

Pri skenovaní ste však prstom hýbali, a teda každá snímka môže byť posunutá o nejaký neznámy (celočíselný) počet stĺpcov. V tom prípade snímač na stĺpcoch, ktoré sa neprekrývali s prstom, určite nezaznamenal žiadne hrbolčeky.

Snímač navyše sníma iba ak vidí aspoň jeden hrbolček, a teda každá snímka ho bude obsahovať aspoň jeden (napríklad snímka plná políčok 3. typu nebude na vstupe).

Vy samozrejme svoj odtlačok prsta nepoznáte. Ste posadnutý zistením, či ste vy neschopní, softvér hlúpy alebo snímač pokazený:

1. Ak existuje poposúvanie prsta, ktoré je konzistentné so všetkými snímkami a toto poposúvanie určuje práve jeden výzor prsta, tak je softvér hlúpy.
2. Ak existuje poposúvanie prsta, ktoré je konzistentné so všetkými snímkami, ale žiadne poposúvanie nevie jednoznačne určiť, ako vyzerá prst, tak ste vy neschopní.
3. Ak neexistuje poposúvanie prsta, ktoré je konzistentné so všetkými snímkami, tak je snímač pokazený.

Odporúčame vám si tieto pravidlá prečítať ešte raz. Zdôrazneniavhodné je napríklad to, že softvér môže byť hlúpy aj ak existuje niekoľko rôznych odtlačkov prsta, z ktorých mohli vzniknúť naše snímky.

Vstup a Výstup

Na prvom riadku vstupu budú tri čísla n_x , n_y a k ($1 \leq n_x, n_y, k \leq 10$) udávajúce počet stĺpcov, riadkov a snímkov. Snímky budú na vstupe vedľa seba oddelené medzerami. Nasleduje n_y riadkov, každý obsahuje $k \cdot (n_x + 1)$ znakov (vrátane znaku '\n'). Snímky obsahujú iba znaky #?., kde # je hrbolček, . je nehrbolček a ? je neistota.

Na výstup vypíšte jeden riadok obsahujúci jedno slovo **hlupy**, **neschopny** alebo **pokazeny** podľa toho či je softvér hlúpy, vy neschopní alebo snímač pokazený.

Príklady

vstup

```
3 3 1
.#
.##
###
```

Snímač perfektne zachytil snímku odtlačku prsta.

výstup

```
hlupy
```

vstup

```
3 3 3
.# ..? .#.
.## ..# ##?
### .## ##.
```

Odtlačok prsta vyzerá rovnako ako prvá snímka.

výstup

```
hlupy
```

vstup

```
3 3 3
.# .# .??
.# #.. .??
... .. ???
```

Odtlačok prsta vyzerá rovnako ako prvá snímka.

výstup

```
hlupy
```

vstup

```
3 3 3
.# .# .#
?# #.. .?#
... .. .
```

Nie je možné určiť ako vyzeral váš odtlačok prsta.

výstup

```
neschopny
```

vstup

```
3 3 3
.# .# ???
.# #.. ???
... .. ?#?
```

Prvá a tretia snímka si protirečia.

výstup

```
pokazeny
```

vstup

```
2 1 1
.#
```

Ak sme prst nasníмали naozaj celý (neposunuli sme ho), tak ho máme celý jednoznačne určený.

výstup

```
hlupy
```

vstup

```
2 1 2
.# ##
```

V druhom snímku bol prst neposunutý, v prvom posunutý o stĺpec doľava.

výstup

```
hlupy
```

J: Javor do súťaže

80 bodov

Je to tu. Jak každý rok. Jedinečná príležitosť presláviť sa po celom svete: súťaž KSP (Kráľovia Stromových Pestovateľov). Jednoznačne najlepší spôsob, ako svoj obchodík so záhradníckymi pomôckami presláviť na trhu.

Janka o víťaztve snívala hádam už od... jedenástych narodenín. Jednosmerný lístok ku sláve jej rástol pod oknom jednoizbového bytu: Juhoplodný Jantárovolistý Javor, ktorý vypestovala, nemá na svete konkurencie.

Ježiši Mária! Jeleň jej ho obhrýzol. Jój, už vôbec nevyzerá tak pekne, ako mal.

Javí sa jediná možnosť - bude musieť strom obstrihať. Jedna vec je istá - nemôže to spraviť len tak ledabolo.

Jasné! Jablň, ktorá minulý rok vyhrala súťaž, mala zaujímavý štýl, môžem ho napodobniť, pomyslela si. Jej pomocníci dorazili akonáhle si prečítali jej žiadosť o pomoc: junáci Jakub, Jozef a Jaroslav.

Javí sa však, že obstrihať strom na štýl minuloročnej jablone sa dá veľa spôsobmi, a kým sa na to Janka a jej pomocníci podujmú, chceli by vedieť, koľko majú možností.

Jasajte! Jednotlivec, ktorý toto musí zistiť, ste vy. Juchú!

Úloha

Janka má Juhoplodný Jantárovolistý Javor, inak povedané strom: graf, v ktorom sa z každého vrcholu dá dostať do každého iného práve jednou postupnosťou hrán. Tento strom je samozrejme zakorenený.

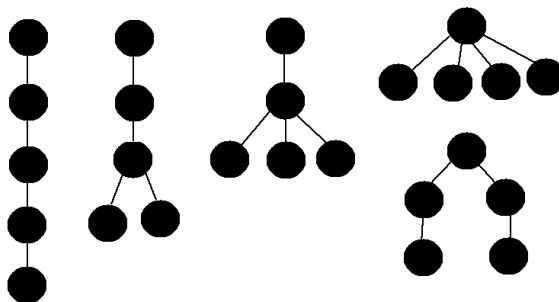
Potrebuje ho obstrihať tak, aby mal nasledovnú vlastnosť: ak sú vrcholy A a B rovnako hlboko v strome (rovnako vzdialené od koreňa), tak musí platiť, že A aj B majú rovnako veľa priamych potomkov.

Ešte si nieje istá, ako ho obstrihá, tak ju zaujíma, koľko rôznych stromov má na výber pre rôzne veľkosti (počty vrcholov).

Zrátajte, koľko stromov o n vrcholoch má vyššie popísanú vlastnosť.

Dva stromy považujeme za rovnaké, ak existuje bijekcia medzi ich vrcholmi (vieme prečíslovať vrcholy prvého stromu tak, aby podstromy každého vrcholu boli rovnaké ako v druhom strome).

Pre ujasnenie znázorníme všetky rôzne stromy s piatimi vrcholmi:



Keďže toto číslo môže byť obrovské, vypíšte tento počet modulo $10^9 + 7$.

Vstup a Výstup

V prvom riadku je číslo $1 \leq t \leq 1000$ - počet rôznych počtov vrcholov, ktoré Janka zvažuje.

V každom z nasledujúcich t riadkov je jedno číslo n : počet vrcholov.

Pre každé n na vstupe vypíšte jedno číslo: koľko existuje stromov o n vrcholoch, kde rovnako hlboké vrcholy majú rovnako veľa priamych potomkov.

V prvej sade $1 \leq n \leq 6$. V druhej sade $1 \leq n \leq 40$. V tretej sade $1 \leq n \leq 1000$. V štvrtej $1 \leq n \leq 10^5$.

Príklad

vstup

```
3
1
2
5
```

výstup

```
1
1
5
```

Pre $n = 1$ existuje jediný strom: sám koreň. Pre $n = 2$ koreň s jedným potomkom. Pre $n = 5$ sú možnosti znázornené v obrázku v zadaní.