Forritunarkeppni Framhaldsskólanna 2013

Spock deild - fyrir hádegi

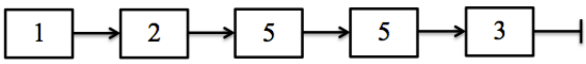
Háskólinn í Reykjavík

16. mars 2013

Verkefni 1 – Tengdur Listi

Gagnaskipan sem oft er notað í Tölvunarfræði kallast Tengdur listi. Hann er notaður til að geyma runu af stökum (líkt og fylki), en að framkvæma mismunandi aðgerðir (eins og að bæta við staki, fjarlægja stak, eða ná í stak á ákveðnum stað) getur verið bæði hægara eða hraðara en á venjulegu fylki sem inniheldur sömu stök. Tengdur listi er ekki einungis gagnlegur einn og sér, heldur er hann líka grunnurinn að mörgum flóknari gagnaskipan.

Tengdur listi samanstendur af svokölluðum hnútum. Hver hnútur geymir ákveðið stak í listanum, og bendir svo á næsta hnút. Síðasti hnúturinn í listanum geymir síðasta stakið í listanum, og bendir svo ekki neitt, eða hefur svo kallaðan „*NULL*“ bendi.



Tökum sem dæmi tengdan lista með stökunum $5, 8, 5$ og 7 í þeirri röð. Þá er einn hnútur fyrir hvert stak, og hver hnútur bendir á þann næsta. Síðasti hnúturinn hefur svo „*NULL*“ bendi. Tengdi listinn fyrir stökin að ofan má tákna á eftirfarandi hátt:

```
[5]->[8]->[5]->[7]->NULL
```

Ein aðgerð sem oft þarf að gera á tengan lista er að snúa honum við. Til dæmis, ef við snúum listanum að ofan við, þá fáum við eftirfarandi tengdan lista:

```
[7]->[5]->[8]->[5]->NULL
```

Þú átt að lesa inn tengdan lista sem er táknaður á þessu formi, snúa honum við, og skrifa hann út á sama formi.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda tengdra lista til að snúa við. Svo fylgja T línur, þar sem hver lína er tengdur listi á forminu sem sýnt er að ofan. Gera má ráð fyrir að hver tengdur listi innihaldi að minnsta kosti eitt stak, og að hvert stak sé heiltala á bilinu 0 til 10000.

Úttak

Fyrir hvern tengdan lista í inntakinu á að skrifa út eina línu sem inniheldur tengda listann þegar búið er að snúa honum við, líka á forminu sem sýnt er að ofan.

Dæmi

Inn	Út
5 [5]->[8]->[5]->[7]->NULL [1]->[2]->[3]->NULL [0]->[150]->[300]->NULL [16]->[8]->[16]->NULL [1337]->NULL	[7]->[5]->[8]->[5]->NULL [3]->[2]->[1]->NULL [300]->[150]->[0]->NULL [16]->[8]->[16]->NULL [1337]->NULL

Verkefni 2 – Óendanleg Runa

Jón litli á von á vini sínum í heimsókn, en þeir eru búnir að skipuleggja brjálað LAN kvöld og ætla að spila alla uppáhalds tölvuleikina sína. Jón er búinn að bíða í allan dag og er ekkert smá spenntur. En þá hringir vinur hans og segist því miður ekki geta komist, en hann sé orðinn fárveikur.

Það er ekkert gaman að spila þessa tölvuleiki einn, þannig Jóni fer að leiðast. Þá fær hann allt í einu snilldar hugmynd. Hann ætlar að búa til óendanlega stóra tölu með því að taka eitthverja litla tölu og bæta henni aftan við sig óendanlega oft.

Til dæmis, hann velur töluna 6883, og bætir henni óendanlega oft aftan við sig, og fær þá óendanlega stóru töluna

$$68\underline{83688}3688368836883688368836883688368836883688368836883688368836883\ldots$$

Hann byrjar að skrifa töluna niður á blað, en eftir að hafa skrifað í hálftíma og fyllt tvær heilar blaðsíður ákveður hann að stoppa. Talan er svo stór! Hún er óendanlega stór! Honum langar að skoða töluna betur, og þá sérstaklega hvernig smá partur af henni lítur út á ákveðnum stöðum. Hann ákveður því að búa til forrit sem les inn upphaflegu töluna sem notuð var til að búa til óendanlega stóru töluna, og svo tvær heiltölur i og j. Forritið á svo að skrifa út alla tölustafina frá tölustafi númer i í tölunni til tölustafs númer j í tölunni.

En þá allt í einu fattar Jón litli að hann kann ekki að forrita! Hann biður þig því um aðstoð.

Ef við tökum sem dæmi óendanlega stóra töluna sem búin er til með tölunni 6883, og við viljum fá bútinn af tölunni frá $i = 3$ og upp að $j = 7$, þá mun forritið skila 83688. Búturinn er undirstrikaður í tölunni að ofan.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af tveimur línum.

Á fyrri línunni eru heiltölurnar i og j, aðskildar með bili, þar sem $1 \leq i \leq j \leq 10^9$ og $j - i < 10^4$.

Á seinni línunni er talan sem notuð var til að búa til óendanlega stóru töluna. Athuga skal að þessi tala getur verið allt að 100 tölustafir að lengd, en hefur að minnsta kosti einn tölustaf.

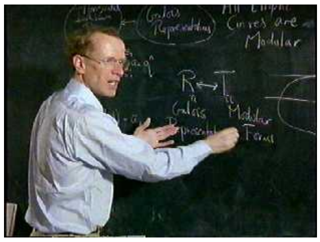
Úttak

Skrifa skal út eina línu með bútinum af óendanlega stóru tölunni frá tölustafi númer i til tölustafs númer j. Athuga skal að fyrsti tölustafurinn er númer 1, annar tölustafurinn númer 2, og svo framvegis.

Dæmi

Inn	Út
4 3 7 6883 10 15 123 1 1 9124232342825 5000000 5000004 3665333693	83688 123123 9 33665

Verkefni 3 – Fermat's Last Theorem



Pierre de Fermat er franskur stærðfræðingur sem uppi var á 17. öld. Þekktastur er hann fyrir eftirfarandi tilgátu:

> *Ekki eru til neinar pósitífar heiltölur a, b og c sem uppfylla jöfnuna $a^n + b^n = c^n$ þegar n er heiltala stærri en 2.*

Fermat skrifaði þessa tilgátu í spássíu bókar sem hann var að lesa, og sagðist hafa sönnun fyrir tilgátunni, en hún væri of löng til að komast fyrir á spássíunni. Margir stærðfræðingar reyndu að sanna tilgátuna eftir þetta, en það var ekki fyrr en rúmlega 300 árum síðar að breski stærðfræðingurinn Andrew Wiles kom loksins með sönnun. Tilgátan reyndist því vera sönn, og er nú kölluð „síðasta setning Fermats."

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu með heiltölunni $2 < n < 10^{100}$.

Úttak

Fyrir hvert prófunartilvik á að skrifa út allar pósitífar heiltölur a, b og c sem uppfylla jöfnuna $a^n + b^n = c^n$. Skrifa á út eina lausn á línu á forminu „$a \ b \ c$". Ekki skiptir máli í hvaða röð lausnirnar eru skrifaðar út. Ef það eru ekki neinar lausnir, þá á að skrifa út eina línu á forminu „`no solutions when n =` n".

Dæmi

Inn	Út

`2`
`3`
`42`
	`no solutions when n = 3`
	`no solutions when n = 42`

Verkefni 4 – Sierpinski Triangle

Pascal-þríhyrningurinn er óendanlega stór þríhyrningur af heiltölum, skilgreindur þannig að efst í þríhyrningnum er talan 1, og er talan 1 svo í fyrsta og síðasta dálki hverrar raðar. Tölur annarstaðar í þríhyrningnum eru fengnar með því að leggja saman tölurnar tvær fyrir ofan hana. Fyrstu sjö raðirnar í þríhyrningnum líta þá svona út:

$$
\begin{array}{ccccccccccccc}
 & & & & & & \boxed{1} & & & & & & \\
 & & & & & \boxed{1} & & \boxed{1} & & & & & \\
 & & & & \boxed{1} & & 2 & & \boxed{1} & & & & \\
 & & & \boxed{1} & & \boxed{3} & & \boxed{3} & & \boxed{1} & & & \\
 & & \boxed{1} & & 4 & & 6 & & 4 & & \boxed{1} & & \\
 & \boxed{1} & & \boxed{5} & & 10 & & 10 & & \boxed{5} & & \boxed{1} & \\
\boxed{1} & & 6 & & \boxed{15} & & 20 & & \boxed{15} & & 6 & & \boxed{1}
\end{array}
$$

Ef við litum reitina sem innihalda oddatölur svarta, og reitina sem innihalda sléttar tölur hvíta, þá fáum við mynstur. Þetta mynstur heitir Sierpinski þríhyrningurinn, og er svokallaður „fractal“.

Inntak

Inntakið inniheldur eina heiltölu $1 \leq n \leq 100$.

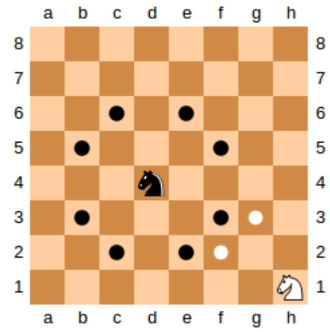
Úttak

Skrifa á út fyrstu n raðirnar í Sierpinski þríhyrningnum, þar sem svartur dálkur er táknaður með '#', og hvítur dálkur er táknaður með '.'. Athuga skal að þríhyrningurinn á að halla til vinstri.

Dæmi

Inn	Út
16	# ## #.# #### #...# ##..## #.#.#.# ######## #.......# ##......## #.#.....#.# ####....#### #...#...#...# ##..##..##..## #.#.#.#.#.#.#.# ################

Verkefni 5 – Konunglegur Matur



Hvíti kóngurinn og hvíta drottningin í skáklandi voru að fara að undirbúa kvöldmáltíðina þegar þau taka eftir því að það vantar sósu til að hafa með matnum. Venjulega myndu þau láta riddara skreppa út í búð fyrir sig, en báðir hvítu riddararnir eru í bardaga við svörtu riddarana, og því ekki heima. Kóngurinn og drottningin ákveða því að annaðhvort þeirra fer út í búð á meðan sá sem bíður heima leggur á borðið. En þau eru bæði orðin mjög svöng, þannig þau vilja að sá sem er fljótari að fara út í búð fari.

Skákland er í rauninni bara skákborð af stærð $N \times N$. Konungsfjölskyldan er staðsett á reitnum (s_r, s_c), á meðan búðin er staðsett á reitnum (t_r, t_c). Þegar kóngurinn og drottningin eru að ferðast, þá taka þau skref. Í einu skrefi getur kóngurinn farið einn áfram í hvaða átt sem er (og þá eru áttirnar á ská líka teknar með), á meðan að drottningin getur í einu skrefið farið eins marga áfram og hún vill í hvaða átt sem er (og þá eru áttirnar á ská líka teknar með). Athuga skal að þessi skref eru alveg eins í hefðbundinni skák fyrir kóng og drottningu. Líka skal athuga að hvorki kóngurinn né drottningin mega fara úr skáklandi þegar þau eru að taka skref.

Skrifið forrit sem les inn N, t_r, t_c, s_r og s_c, og skrifar út hver þarf færri skref til að komast út í búð, og lætur vita ef skrefafjöldi er sá sami fyrir bæði kónginn og drottninguna.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu með heiltölunum $1 \leq N \leq 1000000, 1 \leq t_r, t_c, s_r, s_c \leq N$.

Úttak

Fyrir hvert prófunartilvik á að skrifa út eina línu sem inniheldur „`King`“ ef kóngurinn þarf færri skref til að komast út í búð, „`Queen`“ ef drottningin þarf færri skref til að komast út í búð, en „`Same`“ ef drottningin og kóngurinn þurfa jafn mörg skref til að komast út í búð.

Dæmi

Inn	Út
2 8 4 4 2 2 100 50 50 50 50	Queen Same

Útskýring á dæmi

Í fyrra prófunartilvikinu eru kóngurinn og drottningin staðsett á reitnum $(4, 4)$, en búðin er staðsett á reitnum $(2, 2)$. Þá getur drottningin farið í búðina í einu skrefi, á meðan kóngurinn þarf tvö skref til að fara í búðina.

Í seinna prófunartilvikinu eru kóngurinn og drottningin staðsett á reitnum $(50, 50)$, en það er sami reitur og búðin er á. Bæði drottningin og kóngurinn þurfa því bæði engin skref til að fara í búðina.

Verkefni 6 – Frumtölur

Frumtala er heiltala, stærri en einn, sem hefur hefur enga deila aðra en 1 og sjálfa sig. Til dæmis er heiltalan 8 ekki frumtala þar sem hún hefur deilana 1, 2, 4 og 8, en heiltalan 7 er frumtala þar sem hún hefur aðeins deilana 1 og 7 (þ.e. 1 og sjálfa sig). Frumtölur koma oft upp í strjálli stærðfræði, og eru lykilatriði í nútíma dulmálsfræði.

Ef við teljum fyrstu fimm frumtölurnar upp, þá er fyrsta frumtalan 2, önnur frumtalan 3, þriðja frumtalan 5, fjórða frumtalan 7, og fimmta frumtalan 11. Þetta dæmi snýst um að finna frumtölu númer n. Til dæmis ef $n = 5$, þá viljum við finna fimmtu frumtöluna, sem er auðvitað frumtalan 11.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu með heiltölunni $1 \leq n \leq 1000$.

Úttak

Fyrir hvert prófunartilvik á að skrifa út eina línu sem inniheldur frumtölu númer n.

Dæmi

Inn	Út
4 1 5 15 100	2 11 47 541

Verkefni 7 – Fatahengi Sveins Litla

Sveinn litli hefur ákveðið að stofna leikhús með litlu vinum sínum, Jóni og Palla. Sveinn litli er því miður ekki mikill leikari, en í staðinn fékk hann það spennandi starf að hafa umsjón með fatahenginu. Hans hlutverk er að taka við yfirhöfnum gesta og geyma þær á meðan á sýningu stendur. Eftir sýninguna koma gestirnir og sækja yfirhafnirnar sínar áður en þeir fara.

Í kvöld er nú ansi stórt kvöld fyrir þá félaga, Svein, Jón og Palla, en það munu koma N $(1 \leq N \leq 9)$ gestir að sjá sýninguna þeirra!

Núna eru gestirnir byrjaðir að streyma inn, og hver á eftir öðrum skilur yfirhöfnina sína eftir í öruggum höndum Jóns litla, yfirumsjónarmanns fatahengisins. Þegar allir N gestirnir hafa skilið eftir yfirhöfnina sína og farnir að horfa á sýninguna, þá tekur Jón litli eftir að hann gleymdi að merkja yfirhafnir gestanna. Núna er hann í vanda, því hann veit ekki hvaða gestur á hvaða yfirhöfn. Hann ákveður að láta sem allt sé í lagi, og þegar gestirnir koma að ná í yfirhafnirnar sínar lætur hann hvern gest fá yfirhöfn af handahófi.

Núna eru gestirnir farnir. Jón litli fer nú að hugsa með sér hversu slæmt þetta gæti hafa verið. Hann ákveður að telja fjölda möguleika á því að enginn gestur hafi fengið sína yfirhöfn. Eftir að hafa reynt að telja í dálítinn tíma gefst Jón litli upp. Hann er ekki búinn að læra að telja svona hátt, og biður hann þig um að hjálpa sér.

Skrifið forrit sem telur fjölda möguleika á því að enginn af N gestunum hafi fengið sína yfirhöfn.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 10$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu með heiltölunni $1 \leq N \leq 9$.

Úttak

Fyrir hvert prófunartilvik á að skrifa út eina línu með fjölda möguleika á því að enginn af N gestunum hafi fengið sína yfirhöfn.

Dæmi

Inn	Út
3 1 3 5	0 2 44

Útskýring á dæmi

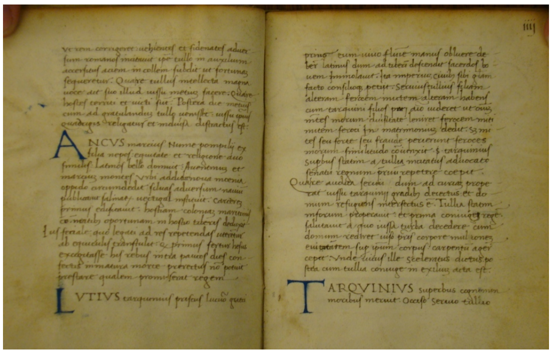
Ef það er bara einn gestur, þá mun hann auðvitað fá sína yfirhöfn þegar hann fer. Þá eru því 0 möguleikar á að enginn fái sína yfirhöfn.

Ef það eru þrír gestir, þá eru tveir möguleikar á að enginn fái sína yfirhöfn:

1. Gestur nr. 1 fær yfirhöfn gests nr. 3, gestur nr. 2 fær yfirhöfn gests nr. 1, og gestur nr. 3 fær yfirhöfn gests nr. 2.

2. Gestur nr. 1 fær yfirhöfn gests nr. 2, gestur nr. 2 fær yfirhöfn gests nr. 3, og gestur nr. 3 fær yfirhöfn gests nr. 1.

Verkefni 8 – Skemmtilegar Setningar



Setning samanstendur af orðum, aðskildum með bili, þar sem hvert orð er röð af bókstöfum. Setning getur svo endað á táknum eins og punkti, spurningamerki, eða upphrópunarmerki. Við skulum kalla setningu *skemmtilega* ef að fyrir hvert par af samliggjandi orðum gildir að síðasti stafurinn í fyrra orðinu er sá sami og fyrsti stafurinn í seinna orðinu. Þá er til dæmis setningin „Ýmir rúllaði inní ísbúðina!“ skemmtileg, en við sjáum að síðasti stafurinn í „Ýmir“ er sá sami og fyrsti stafurinn í „rúllaði“, og það sama gildir fyrir „rúllaði“ og „inní“, og „inní“ og „ísbúðina“. Athuga skal að ekki er gerður greinarmunur á hástaf og lágstaf. Ef setning er ekki *skemmtileg*, þá er hún auðvitað kölluð *leiðinleg*.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu sem inniheldur eina setningu, þar sem setning er skilgreind að ofan.

Úttak

Fyrir hvert prófunartilvik á að skrifa út eina línu sem inniheldur „`Fun`“ ef setningin er skemmtileg, en „`Boring`“ ef setningin er leiðinleg.

Dæmi

Inn	Út

`6`
`Hvernig gengur?`
`Forritunarkeppnin er awesome.`
`Was she early?`
`Abba Abba abba.`
`Abba babba babb.`
`John needs sugar!`
	`Fun`
	`Boring`
	`Fun`
	`Fun`
	`Boring`
	`Fun`

Verkefni 9 – Tic Tac Toe

Í leiknum Tic-Tac-Toe skiptast leikmenn á að setja táknið sitt í reiti á fylki af stærð 3×3. Fyrri leikmaðurinn notar táknað `X`, en seinni leikmaðurinn notar táknið `O`. Leikmaður vinnur þegar hann hefur fyllt heila röð, dálk, eða aðra hvora skálínuna með tákninu sínu. Til dæmis hefur leikmaður með táknið `O` unnið þennan leik:

X	X	
O	O	O
X		

Aftur á móti hefur enginn enn unnið þennan leik:

X		X
X	O	X
O	O	

Líka er hægt að spila Tic-Tac-Toe á stærra borði, eða á fylki af stærð $n \times n$, þar sem $n \geq 3$. Þá vinnur leikmaður ef hann hefur fyllt einhverja röð, einhvern dálk, eða aðra hvora af skálínunum með tákninu sínu.

Skrifið forrit sem les inn heiltöluna n, og svo $n \times n$ fylki með táknunum `X` og `O`. Forritið á svo að skrifa út hvort leikmaður `X` sé búinn að vinna, leikmaður `O` sé búinn að vinna, eða enginn sé enn búinn að vinna. Ef sú staða kemur upp að báðir leikmenn hafa unnið á að skrifa út villumeldingu.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 100$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik byrjar á einni línu með heiltölunni $3 \leq n \leq 100$. Næstu n línur innihalda hver n stafi, þar sem stafirnir eru `X`, `O`, og `.` (punktur). Punktur táknar auðan reit.

Úttak

Fyrir hvert prófunartilvik á að skrifa út eina línu sem inniheldur „`X won`“ ef leikmaður `X` vann, „`O won`“ ef leikmaður `O` vann, „`Neither won`“ ef enginn er enn búinn að vinna, eða „`Error`“ ef báðir eru búnir að vinna.

Dæmi

Inn	Út
4 3 .OX OOX X.X 3 O.O XOX XXO 5 O...X O...X O...X O...X O...X 4 ...O ..OX ..XX O..X	X won O won Error Neither won

Verkefni 10 – Bitmask

Bitagrímur (eða *bitmasks*) koma oft upp þegar unnið er með bita og bitareikning. Með þessum grímum er hægt að fá fram allskonar bitamynstur. Til dæmis er hægt að núllstilla fyrstu fjóra bitana í 32-bita tölu með því að nota bitagrímuna `0xFFFFFFF0` og nota svo `AND` bitaaðgerðina. Í þessu dæmi átt þú að finna svona bitagrímu.

Þú færð gefnar þrjár 32-bita jákvæðar heiltölu N, L og U. Þú átt að finna bitagrímu M þannig að $L \leq M \leq U$ og N `OR` M er í hámarki. Til dæmis ef N er 100, og $L = 50$, $U = 60$, þá mun M vera 59 og N `OR` M mun vera 127, sem er í hámarki. Ef fleiri en eitt gildi fyrir M uppfylla þessi skilyrði, þá skaltu nota minnsta gildið.

Inntak

Á fyrstu línu er heiltalan $1 \leq T \leq 1000$, sem táknar fjölda prófunartilvika sem fylgja. Hvert prófunartilvik samanstendur af einni línu með þremur 32-bita jákvæðum heiltölum N, L og U, þar sem $L \leq U$.

Úttak

Fyrir hvert prófunartilvik á að skrifa út minnsta gildi fyrir M þannig að N `OR` M er í hámarki.

Dæmi

Inn	Út
5 100 50 60 100 50 50 100 0 100 1 0 100 15 1 15	59 50 27 100 1

Athugið

Látum x og y vera 32-bita pósitífar heiltölur. Í forritunarmálunum `C++`, `Python` og `Java` er bitaaðgerðin x `AND` y táknuð x `&` y, og bitaaðgerðin x `OR` y táknuð x `|` y.