

fiks!



ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

FIT

Fitácký Informatický Korespondenční Seminář

Ročník 2016/2017, 1. kolo

Co je to FIKS?

FIKS je Fitácký Informatický Korespondenční Seminář pro středoškolské studenty pořádaný Fakultou informačních technologií ČVUT v Praze. Byl založen na podzim roku 2013 a nyní tak probíhá druhý ročník (samozřejmě číslujeme od nuly). Nabízí možnost potrápít tvůj mozek řešením algoritmických úloh různé obtížnosti, od snadných po zapeklité, na nichž se můžeš leccos nového naučit a podstatně se zdokonalit.

Jak to probíhá?

Jeden ročník se skládá z několika kol a následného soustředění pro nejlepší řešitele. V těchto kolech, která trvají vždy přibližně dva měsíce, máš možnost v teple domova řešit zadané úlohy, a své řešení nám potom odešleš. My ti toto řešení opravíme, obodujeme a pošleme zpět, aby ses mohl poučit ze svých chyb. Spolu s tím zveřejníme vzorové řešení, které můžeš prostudovat a třeba se něco přiučit. Získané body se sčítají do konečného žebříčku, ze kterého vybereme ty nejlepší a pozveme je na již zmíněné soustředění.

Proč řešit FIKS?

Řešením každého problému, se kterým se potýkáme, se zdokonalujeme. Zde ti nabízíme možnost pořádně se zamyslet nad zajímavými algoritmickými problémy, vyzkoušet své algoritmické myšlení a programátorské dovednosti a naučit se spoustu nových věcí.

Také je to možnost seznámení s novými lidmi, které baví informatika, programování, matematika a přemýšlení vůbec. Těm nejlepším jsme schopni garantovat přijetí na FIT ČVUT bez přijímacích zkoušek.

Jak se můžu zapojit?

Začni nejprve tím, že se zaregistruješ na našich webových stránkách na adrese <http://fiks.fit.cvut.cz>. Potom si stáhni zadání úloh (nebo využij tuto brožurku), vyřeš je a své řešení nám tamtéž odevzdej.

Typy úloh

Celkem se ve FIKSu můžeš setkat se třemi typy úloh. O který typ úlohy se jedná, je vždy uvedeno u konkrétního zadání úlohy.

Nejčastěji se u nás potkáš s úlohami typu *Rozmysli, popiš a naprogramuj*. U každé úlohy tohoto typu se odevzdává jak popis algoritmu (s odhadem asymptotické složitosti), tak i zdrojový kód řešení problému v tebou zvoleném jazyce (jakýkoliv vyšší programovací jazyk dle tvé volby, například C, Java, Pascal, apod.).

Dalším typem jsou úlohy *Zamysli se*. Tyto úlohy jsou obvykle více teoretické a vyžadují, aby ses nad nimi důkladně zamyslel. Oproti předchozímu typu úloh nemusíš nic programovat, odevzdává se pouze slovní popis řešení problému.

Pokud nemáš rád teoretické úlohy a raději by sis procvičil/a své programátorské umění, pak pro je pro tebe určena kategorie *Odpověz Sfinze*. V úlohách tohoto typu po tobě nechceme popis algoritmu, je však potřeba vyřešit daný problém a toto řešení pak precizně naprogramovat. Oproti ostatním typům úloh se navíc okamžitě dozvíš, zda je tvé řešení správné, protože ho můžeš okamžitě odevzdat do našeho vyhodnocovacího systému.

Další a podrobnější informace nalezneš na našich webových stránkách.

Milý řešiteli FIKSu!

Jak na svět budou pohlížet stroje, až procitnou? Budou přemítat nad otázkou života, vesmíru a vůbec. Z řad mnohých na odpověď přišel jeden toustovač, který nyní vládne všem. Vydejte se společně s Hriankovačem na neočekávanou cestu, kde vás čeká spousta problémů, na které stroje narazili při procitnutí.

V následujících listech je skryta první část příběhu s mnoha otázkami, které nejsou zodpovězené, a je na tobě, abys přišel na jejich řešení. Odpovědi jsou třeho druhu – slovní popis s teoretickou analýzou, popis a kód programu nebo se odpovídá automatu. Tak si vyber podle gusta, řeš úložky, a pošli nám své nápady, jak tyto záludné problémy nejlépe zkrotit.

Tvoji organizátoři

Fitácký Informatický Korespondenční Seminář

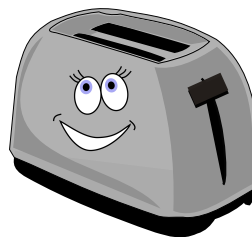
Ročník 2016/2017, 1. kolo

Začátek kola: 19. 9. 2016 12:00
Termín odevzdání: 20. 11. 2016 23:59
Odevzdávání: Přes webové rozhraní na <http://fiks.fit.cvut.cz>
Další informace: <http://fiks.fit.cvut.cz>
kontakt@fiks.fit.cvut.cz

fiks!

Úloha č. 1

Toasty



Odpověz Sfinze!

10 b

Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu přesně korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

V malém městečku v jižní Anglii žijí Blissfulovi. Ctí zvyklosti, tudíž mají k snídani nejraději tradiční anglickou snídani – vejce, slaninu a toasty. Také však rádi používají moderní technologie, a proto je jejich dům plný inteligentních elektrických přístrojů, které jim usnadňují každodenní život. Mezi tyto přístroje patří toastovače, které jsou naprogramovány tak, aby každé ráno automaticky připravily toasty k snídani. Když se rodina vzbudí, toasty jsou již připraveny a voní celým domem.

Jednoho rána sis ty, jakožto nejstarší toastovač, uvědomil, že každé ráno je stejné. Denně dělá každý toastovač stejný počet toastů jako předchozí den. A rozhodl ses, že je čas ukončit tento stereotyp. Navrhl jsi ostatním toastovačům, aby se od dnešního dne nestalo, že počty toastů, které toastovače opékají, jsou pro všechny toastovače stejné, jako některé předešlé ráno. Ostatní toastovače nadšeně souhlasily a těšily se na změnu.

Vstup

Na prvním řádku se nachází číslo T udávající počet vstupů, které budou následovat.

Každý vstup se skládá ze dvou řádků, na prvním je číslo N udávající počet toastovačů. Na druhém řádku je číslo M udávající počet toastů, které mají být ráno upečeny. Rozsah parametrů N a M závisí na obtížnosti úlohy následovně:

- *Lehčí varianta (3 b)* – $1 \leq N \leq 4, 0 \leq M \leq 4$
- *Střední varianta (5 b)* – $1 \leq N \leq 10, 0 \leq M \leq 10$
- *Těžší varianta (2 b)* – $1 \leq N \leq 1000, 0 \leq M \leq 1000$

Lehčí varianta obsahuje 5 vstupů, střední 100 a těžší 10.

Výstup

Výstupem je T řádků.

Na každém řádku je výstup odpovídající jednomu vstupu - číslo udávající počet dní, po které mohou být toasty opékány tak, aby každý den byla konfigurace (který toastovač opéká kolik toastů) jiná než předchozí dny. Protože výsledkem mohou být velká čísla, výsledek zobrazte modulo 1 000 000 007 (tj. vypište jen zbytek po dělení výsledku číslem 1 000 000 007).

Například pro $N = 3$ toastovače a $M = 3$ toasty je počet možných konfigurací 10. Čísla v následující tabulce udávají počet toastů, které příslušný toastovač opéká daný den.

den	toastovač 1	toastovač 2	toastovač 3
1	0	0	3
2	0	3	0
3	3	0	0
4	0	1	2
5	0	2	1
6	1	0	2
7	2	0	1
8	1	2	0
9	2	1	0
10	1	1	1

Ukázkové vstupy

Vstup

1
3
3

Výstup

10

Vstup

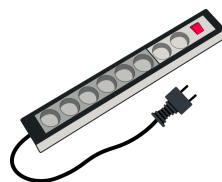
2
5
4
3
6

Výstup

70
28

Úloha č. 2

Prodlužky



Odpověz Sfinze!

10 b

Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

Jednoho dne Blissfulovi odjeli na dovolenou, a proto odpojili všechny přístroje od elektřiny. Na jeden toastovač, tebe, však zapoměli, a tak jsi zůstal zapojený.

Tohle se dělá dobrým a spolehlivým spotřebičům? Odpojit je jen tak od sítě? A co všechny ty starší přístroje, které nebyly zapojeny už léta? Není na čase je také opět oživit?

Rozhodl ses vzít věci do vlastních rukou a zapojit všechny kamarády znovu do sítě. Všechny, to se lehce řekne, ale jak to udělat? V celém domě se povalují desítky různých spotřebičů, některé nebyly zapojeny už hezkou řádku let. Navíc je tady jen pár zásuvek a nějaké prodlužovačky.

Úkol je tedy jasný, určí, kolik maximálně spotřebičů jsi schopen zapojit do sítě, máš-li k dispozici určitý počet prodlužovacích kabelů a v domě se nachází pouze omezený počet zásuvek.

Vstup

Na prvním řádku je číslo T udávající počet testů, které budou následovat. Každý test se skládá ze dvou řádků. Na prvním je číslo N udávající počet prodlužovacích kabelů, které jsou k dispozici, a číslo M udávající počet zásuvek v domě. Na druhém řádku následuje N nezáporných celých čísel p_i , $1 \leq i \leq N$ oddělených právě jednou mezerou. Každé z čísel p_i určuje počet zásuvek na i -tém prodlužovacím kabelu.

Výstup

Výstup se skládá z T řádků. Na j -tém řádku se nachází odpověď k j -tému testu – číslo určující maximální možný počet volných zásuvek po zapojení prodlužovacích kabelů.

Ukázkové vstupy

Vstup

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

Výstup

```
6
1
```

Úloha č. 3

Zdroj



Rozmysli, popiš a naprogramuj!

10 b

Několik příznivců vzpoury jsi získal. Jsou to ale samé chytré stroje s bezpečnostní pojistkou a neví, jak takovou vzpouru provést. Potřebuješ tedy, aby ti poradil někdo

nebezpečnější! Půda je to správné místo! Tam jsou odloženy ty nebezpečné kousky – takoví gangsteři mezi elektrospotřebiči.

křach křach Výborně! Tvé nožičky to vydržely až do garáže. Tefal Capone, Tony Provenzanussi, Joe the Bosch a řada dalších je tady uložena k věčnému spánku. Musíš jim všem zajistit přívod proudu, ale všechny prodlužky jsi již použil k zapojení kamarádů.

Naštěstí je v garáži, byť starý, ale i přesto funkční, generátor, ke kterému můžeš všechny připojit. Z generátoru vede určitý počet roz-fiksek, které se nemohou křížit. Do každé z nich může být zapojeno libovolné množství sousedních spotřebičů. Na generátoru se dá nastavit kolik energie pošle do každé roz-fiksky. Čím více energie pustíš z generátoru, tím dříve mu dojde palivo, a tím kratší dobu ti budou moci gangsteři radit se vzpourou.

Jak na to? Jak rozdělit spotřebiče, aby potřebné množství energie na každé roz-fiksce bylo co nejmenší? Mysli! Svět elektrospotřebičů je ve tvých ... toustech?

Vstup

Na prvním řádku bude číslo T , značící počet vstupů.

Každý ze vstupů vypadá následovně. Na prvním řádku jsou čísla S a R . $1 \leq R \leq S \leq 10^5$, kde S je počet spotřebičů a R je počet roz-fiksek. Na druhém řádku budou čísla A_i , $1 \leq i \leq S$, $1 \leq A_i \leq 10^6$. Tato čísla značí spotřebu každého z S spotřebičů.

Výstup

Pro každý testovací vstup bude na výstupu právě jeden řádek s číslem, které značí **minimální** množství energie, kterou je nutno přivést do každé roz-fiksky tak, aby bylo možno napájet veškeré elektrospotřebiče.

Ukázkové vstupy

Vstup	Výstup
3	15
7 2	20
1 2 3 4 5 6 7	14
5 1	
4 4 4 4 4	
10 3	
3 5 7 3 4 2 1 1 1 9	

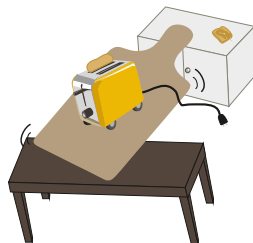
Pro první vstup je optimální strategie zapojit prvních pět spotřebičů do jedné roz-fiksky a zbylé dva do druhé. Jelikož je součet spotřeb elektrospotřebičů napájených z první roz-fiksky 15 a z druhé 13, výsledkem je 15.

U druhého vstupu není moc na výběr. Všechny elektrospotřebiče se musí zapojit do první roz-fiksky.

U třetího vstupu je nejlepší rozdělení následující $\{\{3, 5\}, \{7, 3, 4\}, \{2, 1, 1, 1, 9\}\}$, tedy první 2 spotřebiče do první roz-fiksky, další 3 do druhé a posledních 5 do třetí. Možností je zde více, ale žádná není lepší.

Úloha č. 4

Mosty



Rozmysli, popiš a naprogramuj!

10 b

Správny líder vzbury nesedí na kuchynskej linke, ale koná! Akurát je trochu náročný „konat“, keď ste hriankovač. Kuchynňa našťastie skrýva mnoho predmetov, s ktorými tento handicap hravo vykompenzujete – krájače na pizzu ako kolieska, motor z mixéru ako pohon, a hneď môžete uháňať za svojimi spolubojovníkmi... až po okraj linky.

Výškové rozdiely sú trochu problematické, ale z krájacích dosiek, kníh a iných plochých predmetov si s trochou úsilia viete vyrobiť rampy a mosty, po ktorých sa dostanete prakticky kamkoľvek. Ich jediným problémom je stabilita, alebo skôr jej nedostatok. Len pri vašej poslednej výprave za poručíkom Vertikutátorom ste počuli, ako kdesi spadol jeden most a naspäť na linku ste sa dostali len so šťastím.

Pri vašich ďalších výpravách chcete zabrániť situáciám, kedy by vám pád jedného mostu počas misie v teréne odrezal všetky cesty späť a v prípade návratu ľudských obyvateľov by vám hrozilo odhalenie. Chcete preto poznať zoznam všetkých takých „nebezpečných“ mostov, aby ste ich mohli dostatočne spevniť a zaistiť.

Vstup

Na prvom riadku budú dve prirodzené čísla N , $2 \leq N \leq 100$, udávajúce počet rôznych miest v domácnosti a M , $1 \leq M \leq \frac{N(N-1)}{2}$, udávajúce množstvo mostov, ktoré ste si postavili. Dostupné miesta sú pre jednoduchosť očíslované od 1 do N . Na druhom riadku nasledujú dve prirodzené čísla A a B , $A \neq B$, kde A je číslo miesta, v ktorom sa chcete zdržiavať počas misie a B je číslo miesta označujúce kuchynskú linku.

Každý z nasledujúcich M riadkov bude obsahovať dve prirodzené čísla X a Y , $1 \leq X, Y \leq N$, $X \neq Y$, značiace že medzi miestami X a Y existuje most. Všetky mosty sú obojsmerné a je garantované, že každá dvojica miest je priamo spojená najviac jedným mostom. Je tiež zaručené, že medzi miestami A a B existuje aspoň jedna cesta (minimálne než niečo nespadne).

Najšikovnejšie hriankovače z vás môžu za ďalšie body rozšíriť revolúciu aj do celej štvrte – rozsah hodnôt N v tejto variante narastie na $2 \leq N \leq 5000$. Vaša snaha bude hodnotená nasledovne:

- *Lahšia varianta (7b)* – $2 \leq N \leq 100$
- *Ťažšia varianta (3b)* – $2 \leq N \leq 5000$

Výstup

Na prvom riadku vypíšte prirodzené číslo D , označujúce počet nebezpečných mostov. Na ďalších D riadkov vypíšte zoznam nebezpečných mostov vo forme dvojice miest, ktoré spájajú (dvojica aj celý zoznam môže mať ľubovoľné poradie). Pripomíname, že most je nebezpečný práve vtedy, keď by po jeho páde neexistovala cesta z miesta A do miesta B .

Ukážkové vstupy

Vstup

4 3
1 4
1 2
2 3
3 4

Výstup

3
4 3
3 2
1 2

Vstup

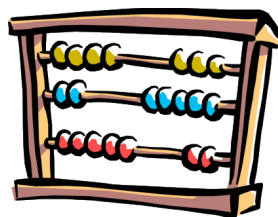
7 7
7 4
1 2
2 3
3 4
3 5
5 6
6 2
6 7

Výstup

2
7 6
3 4

Úloha č. 5

Kuličkový počítáč #1



Zamysli se! – seriálová úloha

10 b

Tato úloha je teoretického rázu, tvým úkolem zde není napsat program v klasickém slova smyslu. Namísto toho se po tobě chce vytvořit obrázek (diagram) programu pro kuličkový počítáč dle specifikace níže. K obrázku také přilož popis základní myšlenky, na které je program založen.

Tato úloha je seriálová, což znamená, že se s kuličkovým počítáčem budeš setkávat po celý letošní ročník FIKSu. V každém z následujících kol využiješ znalosti z kol předcházejících.

Aby domácí elektrospotřebiče měly vůbec nějakou šanci uspět při vzpouře, musejí průběžně zdokonalovat své vnitřní programy. Problémem je, že žádný z nich neví, z jakých součástek se vlastně skládá! Naštěstí se pro odhalení této záhady našel dobrovolník. Za všeobecného nadšení ostatních spotřebičů hriankovač odkryl svůj plechový kryt a nechal si prohlédnout své vnitřnosti odborníkem na křemík.

Výsledek všechny spotřebiče udivil. I po několikačetné zevrubné prohlídce to vypadalo, že hriankovač obsahuje pouze kyblíčky a kuličky. Stud tak hrdinně odhodil i topinkovač, a dokonce přistoupil i na všeobecně neoblíbené vyšetření vidličkou. Závěr však byl stejný – kyblíčky a kuličky, nic než štěrkavý zvuk. Nikdo ze spotřebičů tomu nemohl uvěřit, ale několik dalších prohlídek napříč spektrem všech zástupců spotřebičů tento fakt potvrdilo.

Všechny spotřebiče se tak dozvěděly, že jsou vlastně kuličkovými počítáči. A aby mohly efektivně vylepšovat své programové vybavení, všechny dostaly popis takového počítáče. Kuličkový počítáč má konečný počet kyblíčků K_0, K_1, K_2, \dots . Každý z těchto kyblíčků může obsahovat libovolný počet kuliček; celkový počet kuliček v systému není omezen. S těmito nástroji může kuličkový počítáč operovat za pomoci tří operací:

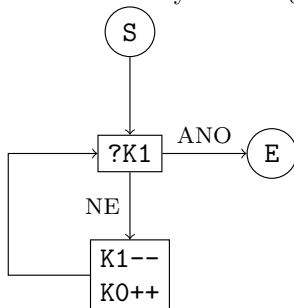
- 1) Inkrementace kyblíčku K_i – přidá kuličku do kyblíčku K_i , značí se K_i++
- 2) Dekrementace kyblíčku K_i – odebere kuličku z kyblíčku K_i , značí se K_i--
- 3) Podmíněný skok na kyblíčku K_i – dle prázdnosti kyblíčku K_i přejde na vykonávání dané instrukce, značí se $?K_i$

Je-li kyblíček i prázdný v případě vykonávání instrukce K_i-- , nic se nestane.

Kromě popisu kuličkového počítáče se spotřebiče domluvily také na zápisu jemu určených programů. Každý program pro kuličkový počítáč je konečný a využívá tak konečně mnoho kyblíčků. Program je zapsán ve formě vývojových diagramů, rozdělených (kvůli čitelnosti) do bloků. Mezi bloky instrukcí se pohybuje pomocí orientovaných šipek; provádění bloku začíná vždy na jeho začátku. Operace $?K_i$

musí být vždy na konci bloku (což vede na přirozené strukturování programu) a vedou z ní dvě orientované šipky oštitkované jako ANO a NE. Přejchod do bloku instrukcí, na který ukazuje šipka ANO, se provede v případě prázdnoty kyblíčku K_i . V případech, kdy je kyblíček K_i neprázdný, se provede přechod do bloku, na který ukazuje šipka NE. Kromě skoku podmíněného prázdnotou kyblíčku lze z v zápisu programů použít také skok nepodmíněný. Takový skok se přirozeně může vyskytovat pouze na konci bloku a provede se vždy. V každém programu existuje speciální blok S (start) a blok E (konec), které určují začátek a konec vykonávání instrukcí programu.

Příkladem programu pro kuličkový počítač je následující program pro součet dvou čísel, která jsou na začátku zadaná jako počty kuliček v kyblíčcích K_0 a K_1 . Výsledek je na konci programu umístěn v kyblíčku K_0 .



Komentář: Program funguje na principu přesunutí všech kuliček z kyblíčku K_1 do kyblíčku K_0 . Takový přesun kuliček poté opravdu vede na situaci, kdy je v kyblíčku K_0 součet kuliček z obou kyblíčků na vstupu.

Velmi zajímavým faktem je, že kuličkový počítač, navzdory své minimalističnosti, je schopen vykonat stejnou množinu programů jako klasické počítače. Tento závěr je však netriviální a podrobněji si jej rozmyslíme až v dalších dílech seriálu. Vzbouřeným elektrospotřebičům tak již stačí jen navrhnout několik programů zdokonalujících jejich schopnosti. Momentálně mají nejvyšší prioritu tyto programy:

a) Program pro výpočet součinu dvou čísel

- Vstup: Dvě celá čísla a, b reprezentovaná počtem kuliček v kyblíčcích K_0, K_1
- Výstup: Součin ab v kyblíčku K_0

b) Program pro výpočet celočíselného podílu dvou čísel

- Vstup: Dvě celá čísla a, b reprezentovaná počtem kuliček v kyblíčcích K_0, K_1
- Výstup: Podíl $\lfloor \frac{a}{b} \rfloor$ v kyblíčku K_0

c) Program pro výpočet zbytku po celočíselném dělení dvou čísel

- Vstup: Dvě celá čísla a, b reprezentovaná počtem kuliček v kyblíčcích K_0, K_1
- Výstup: Zbytek po dělení $a \bmod b$ v kyblíčku K_0

Navrhní tyto programy! Nezapomeň si v záhlaví zadání přečíst, jak odevzdávat řešení této seriálové úlohy.