

fiks!



ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

FIT

Fitácký Informatický Korespondenční Seminář

Ročník 2015/2016, 3. 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!

Milý řešiteli FIKSu! Po všech peripetiích, se kterými ses musel vypořádat, si konečně můžeš trochu oddechnout a dokonce si i užít vánoční atmosféry. Jak jsi totiž zjistil, tak i Martané slaví Vánoce. Nicméně ani na Marsu se Vánoce neobejdou bez problémů, a proto se radě z nich budeš muset podívat na zoubek. Jistě, zachránit celé Vánoce sám se může zdát jako příliš náročný úkol pro jedince, ale když to nedokáže nikdo z Martanů, tak je tato výzva na tobě. Ukážeš všem, že jí dokážeš čelit a zachrániš tak martánské Vánoce?

Volba je na tobě, buď tuhle brožurku odložíš a na FIKS zapomeš, nebo všem ukážeš, že na to máš. Stejně tak jako v prvních dvou kolech najdeš i ve 3. kole FIKSu několik vypečených úloh, nad kterými můžeš za dlouhých zimních večerů trápit svoje mozkové závity. I v případě, že jsi neřešil první dvě kola, tak určitě stojíš za to se zapojit. Navzdory tomu, že někteří účastníci mohou mít bodový náskok z předchozích dvou kol, tak i přesto máš šanci bojovat o ty nejvyšší příčky v celkovém hodnocení!

Součástí tohoto ročníku jsou navíc i tzv. seriálové úlohy, které se věnují jednomu tématu a budou tě provázet od prvního kola do posledního. Vždy se zadáním takové úlohy se dozvíš něco o teorii potřebné k jejímu vyřešení. Během jednoho ročníku se tak do hloubky seznámíš s jedním konkrétním tématem. Aby sis nabyté znalosti mohl dostatečně procvičit, tak můžeš seriálové úlohy řešit kdykoliv, ale s každým dalším kolem se jejich bodové ohodnocení snižuje. Tak neváhej, otoč stránku a pusť se do řešení úloh, které jsme pro tebe připravili!

Tvoji organizátoři

Fitácký Informatický Korespondenční Seminář

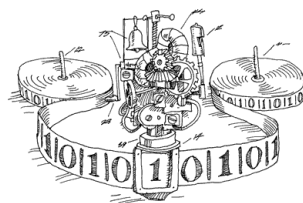
Ročník 2015/2016, 3. kolo

Začátek kola: 14. 12. 2015 00:00
Termín odevzdání: 31. 1. 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. 0a

Lexikální analyzátor #1



Odpověz Sfinze!

2.5/10 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“.

*Tato úloha je součástí seriálu, tzn. že bude otevřena až do konce ročníku, ale každé další kolo se jí zmenší maximální možné ohodnocení na polovinu, proto bys ji měl vyřešit co nejdříve. Pokud jsi tuto úlohu řešil už v minulých kolech, tak ji můžeš s klidem přeskočit. Tato úloha byla zveřejněna v 1. kole, tzn. že její **maximální zisk je hodnocen pouze 25 %**.*

Mise na Mars si žádá to nejmodernější vybavení a přesně takové je k nalezení na vesmírné lodi USS Flexington. Hlavním mozkiem lodi je počítač nejnovější generace schopný provádět i ty nejsložitější výpočty. Aby však počítač mohl něco zpracovávat, je potřeba mu nejprve nějakým způsobem problém zadat. Hlasové ovládání je zatím mimo provoz, inženýrská sekce na opravě pracuje. Kapitán si přeje provést simulaci přistání a k tomu je třeba ověřit základní logické operace. Kvůli opravám bude muset praporečkář Paxson použít řádkový vstup z textové konzole na operačním stanovišti. Napsat program pro zpracování vstupu bude hračka.

Na vstupní řádce jsou celkem tři prvky. Paxson těmto prvkům říká *lexikální jednotky* nebo též *lexémy*. Každý lexém se může skládat z více znaků, praporečkář Paxson tedy musí opatrně zjišťovat, kde končí jeden lexém a začíná jiný. Když jsou všechny lexémy úspěšně načteny, je potřeba jim dodat jejich syntaktický význam. První a třetí lexém jsou operandy, prostřední lexém představuje operátor. Když je i toto zpracované, nic už Paxsonovi nebrání v provedení simulace.

Vstup

Každý vstup na prvním řádku obsahuje celé číslo N . Dále následuje N řádků, každý z nich reprezentuje jedno zadání. Řádek se zadáním obsahuje jeden matematický výraz vyjadřující podmínku. Logický výraz je ve formátu $a \text{ OP } b$, kde a, b jsou celá čísla a OP je jeden z následujících operátorů: $<$, $<=$, $=$, $!=$, $>=$, $>$. Číslo a a operátor jsou vždy odděleny právě jednou mezerou. Maximální rozsah a, b a velikost N jsou $0 < N \leq 1000$ a $-2^{30} \leq a, b \leq 2^{30}$.

Výstup

Výstup obsahuje N řádků. Každý řádek odpovídá jednomu zadání. Obsahem každého řádku je řetězec TRUE nebo FALSE charakterizující logickou hodnotu, která odpovídá vyhodnocení podmínky.

Ukázkové vstupy

Vstup

```
2
1 <= 1
-1 != 1
```

Výstup

```
TRUE
TRUE
```

Vstup

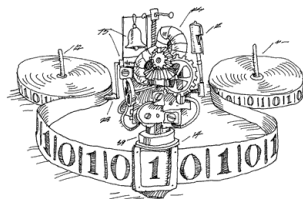
```
3
24 > 24
24 == 24
24 < 24
```

Výstup

```
FALSE
TRUE
FALSE
```

Úloha č. 0b

Lexikální analyzátor #2



Odpověz Sfinze!

5/10 b

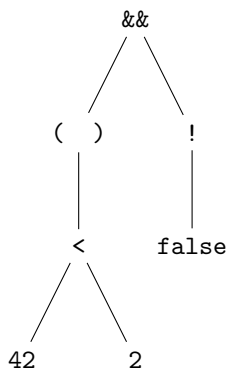
Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu přesně korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

*Tato úloha je součástí seriálu, tzn. že bude otevřena až do konce ročníku, ale každé další kolo se jí zmenší maximální možné ohodnocení na polovinu, proto bys ji měl vyřešit co nejdříve. Pokud jsi tuto úlohu řešil už v minulých kolech, tak ji můžeš s klidem přeskočit. Tato úloha byla zveřejněna v 2. kole, tzn. že její **maximální zisk je hodnocen pouze 50 %**.*

Praporčík Paxton se jistě brzy dočká povýšení za to, jak nedávno naprogramoval řádkový vstup pro simulaci přistávacích manévřů. Operace Mars má velký úspěch a Paxton na tom má velký podíl. Sám Paxton z toho ale tak veselý není. Z povrchu Marsu neustále přicházejí dotazy na hlavní počítač, který se pod tou tíhou zhroutil a praporčík ho musel restartovat. Při tom se ale ztratila důležitá část vstupního

modulu. Loďní počítač přestal načítat a vyhodnocovat logické výrazy. Praporčík ho bude muset napsat celý znovu.

Vstupní řádka obsahuje opět *lexémy*. Tentokrát je však syntaktický význam lexémů složitější. Již se nedá spoléhat na jejich pořadí. Na začátku očekáváme jeden celý výraz. Pokud je na řádce například jen číslo, máme hotovo. Může zde být ale unární operátor (negace), za kterým nutně musí následovat jeden další podvýraz. V případě binárního operátoru budou podvýrazy dva – levý a pravý. Speciálním případem jsou potom závorkové struktury. Otevírací závorka nám vlastně nashodnotuje celý proces znovu, za závorkou očekáváme právě jeden výraz, za kterým musí být závorka uzavírací. Závorky se tak vlastně chovají jako takový „dvojdílný“ unární operátor. Konec řádky se rozpozná jednoduše, přečetli jsme právě tolik podvýrazů, kolik bylo vyžadováno. Celý výraz si tedy můžeme představit jako strom (viz obrázek 0.1).



Obrázek 0.1 Příklad *syntaktického stromu* pro výraz $(42 < 2) \ \&\& \ !\text{false}$

No a jak se takový strom vyhodnocuje? K vyhodnocení libovolného uzlu potřebují znát hodnotu všech jeho potomků. Pokud je uzel listem, jeho hodnotu znám. Stačí tedy začít od listů a postupovat směrem vzhůru ke kořeni.

Vstup

Každý vstup obsahuje N řádků, každý z nich reprezentuje jedno zadání. Řádek se zadáním obsahuje jeden logický výraz.

Logickým výrazem rozumíme následující:

- term,
- term \mathcal{OP} term, kde \mathcal{OP} je jeden z operátorů $\&\&$, $||$, $==$, $!=$,
- numerický výraz.

Pod pojmem term rozumíme:

- číslo a (při vyhodnocování má $a = 0$ logickou hodnotu **false**, jinak **true**),
- logickou hodnotu **true** nebo **false**,
- negaci termu operátorem **!**,

- řádně uzavorkovaný logický výraz.

Numerický výraz je ve formátu $a \mathcal{OP} b$, kde a, b jsou celá čísla a \mathcal{OP} je jeden z následujících operátorů: $<$, \leq , \geq , $>$. Maximální rozsah a, b a velikost N jsou $100 < N \leq 1000$ a $0 \leq a, b \leq 2^{30}$. Vstupní lexémy nemusí být odděleny mezerou.

Výstup

Výstup obsahuje N řádků. Každý řádek odpovídá jednomu zadání. Obsahem každého řádku je řetězec `true` nebo `false` charakterizující logickou hodnotu, která odpovídá vyhodnocení výrazu.

Ukázkové vstupy

Vstup

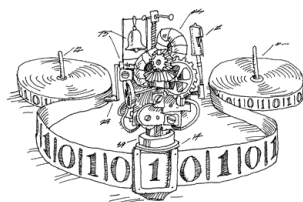
```
true
0
27 < 2
42 || (false)
!false && (23 >= 3)
(true != true) || ((30 > 2) && !(29 <= 1))
true != (!(!!true))
```

Výstup

```
true
false
false
true
true
true
false
```

Úloha č. 1

Lexikální analyzátor #3



Odpověz Sfinze!

10 b

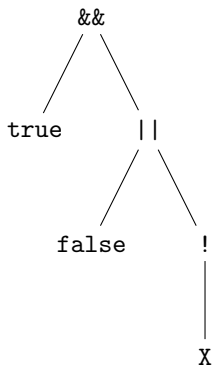
Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu přesně korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

Tato úloha je součástí seriálu, tzn. že bude otevřena až do konce ročníku, ale každé další kolo se jí zmenší maximální možné ohodnocení na polovinu, proto bys ji měl vyřešit co nejdříve.

Program z minulých kol slavil úspěch a praporčík Paxton si užívá zaslouženého odpočinku. Na povrchu se navíc rozmáhá vánoční nálada a tak se na palubě Flingingtonu vůbec nic neděje. Praporčík se rozhodl, že využije volný čas na rozšíření jeho minulého programu. Občas si totiž lidé nejsou při vyhodnocování výrazů jistí všemi hodnotami a tak musí počítač dlouho čekat, dokud není výraz kompletní a

to samozřejmě vyrábí dlouhé fronty u procesoru. Praporčíka napadlo, že pokud by počítač uměl používat proměnné a jen by výrazy zjednodušoval, dalo by se pak jen lehce dosadit a výraz dopočítat, až by byly známé všechny hodnoty.

Na vstupu jsou tedy opět *lexémy*, ze kterých je možné postavit syntaktický strom. V tomto případě máme dva binární operátory, jeden operátor unární a v listech stromu se bude nacházet buď konkrétní hodnota, nebo zástupná proměnná. Závorky jsou tentokrát jen pomocné a určují správnou prioritu operátorů. Příklad takového syntaktického stromu nalezneme na obrázku 1.1.



Obrázek 1.1 Syntaktický strom pro výraz $(\text{true} \ \&\& \ (\text{false} \ || \ !X))$

Vstup

Každý vstup obsahuje N řádků, každý z nich reprezentuje jedno zadání. Řádek se zadáním obsahuje jeden logický výraz.

Logickým výrazem rozumíme následující:

- logickou hodnotu **true** nebo **false**,
- negaci výrazu operátorem **!**,
- řádně uzávorkovaný logický součet,
- řádně uzávorkovaný logický součin,
- název proměnné.

Uzávorkovaný logický součet a součin je ve tvaru $(A \ \mathcal{OP} \ B)$, kde A, B jsou logické výrazy a \mathcal{OP} , je $\&\&$ v případě součinu, nebo $||$ v případě součtu.

Název proměnné je vždy jen jedno velké písmeno anglické abecedy a je zaručeno, že proměnná je v celém výrazu jen jedna a jedenkrát.

Vstupní lexémy nemusí být odděleny mezerou.

Výstup

Výstup obsahuje N řádků. Každý řádek odpovídá jednomu zadání. Obsahem každého řádku je nejkratší možný logický výraz se stejnou pravdivostní tabulkou, jako má výraz vstupní.

Ukázkové vstupy

Vstup

```
true
(true || false)
(!false && X)
(true || (X && false))
!(!true || X)
```

Výstup

```
true
true
X
true
!X
```

Úloha č. 2

Kontrola



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“.

Před dávnými roky, když ještě na Zemi vládli feudální páni, došlo na Marsu k velké stávce pomocných skřítků. Nelíbilo se jim, že mohou na dovolenou pouze v létě, že jsou před Vánoci nuceni pracovat několik měsíců v kuse a nemohou si založit vlastní odbory. Šílenství ohledně skřítků ochromilo celý Mars a společnost rozdělilo na dva nesmiřitelné tábory.

Jedna skupina se bezvýhradně zastávala skřítků a navrhovala, aby byli povzneseni až mezi občany druhé kategorie, zatímco druhá byla striktně proti a za žádných okolností nebyla ochotna ustoupit.

Vyjednávání byla dlouhá a pro zástupce všech stran velice náročná, ovšem skřítkci, zvyklí na dlouhodobý stres a několikátýdenní práci s minimem odpočinku, měli přeci jen navrch.

Nakonec se skřítkům podařilo prosadit několik zákonů, které měly zlepšit podmínky jejich práce. Patrně nejdůležitějším je zákon "O maximální době práce". Je pochopitelné, že se jedná o text velice náročný na čtení a plný právnických výrazů a formulací, který by obyčejný skřítek, jenž se umí sotva podepsat, byl jen stěží s to přečíst. Proto se pro něj vžil spíše název "Zákon 90 na 180", který mnohem lépe vyjadřuje, o co vlastně jde.

Hlavním posláním zákona je omezit vykořisťování skřítků zejména na podzim a v zimě. Bylo tedy ustanoveno, že skřítek smí během každých 180 dnů odpracovat pouze 90, přičemž se počítá každý den, kdy byl skřítek v práci, bez ohledu na počet

odpracovaných hodin. Tedy den, kdy skřítek odpracoval jednu hodinu je z pohledu zákona stejně hodnotný, jako den, kdy odpracoval hodin více.

V továrně, kde se vyrábí tradiční vánoční Coimas, se dozvěděli, že se k nim brzy chystá kontrola z úřadu pro dohled nad zaměstnáváním pomocných skřítků. Rádi by si ověřili, že žádný z jejich zaměstnanců neporušil zmiňovaný zákon, proto tě oslovili, abys jim pomohl napsáním programu, který toto pro každého skřítků zjistí.

Ještě před začátkem práce na programu ses dozvěděl, že stačí ověřovat pouze směny proběhlé od poslední kontroly, která byla provedena přesně před 5 pozemskými léty.

Vstup

Na prvním řádku je číslo T udávající počet skřítků, u kterých je potřeba provést kontrolu odpracovaných hodin.

Každá kontrola pak začíná číslem N představujícím počet směn, které skřítek odsloužil za posledních 5 let. Poté následuje N řádek skládajících se ze dvou čísel a_i a b_i – dnem začátku a konce směny. Dny jsou pro jednoduchost číslovány od poslední kontroly.

Zároveň je zaručeno, že se žádné dvě směny nepřekrývají, tedy žádný z dnů není součástí více než jedné směny.

Výstup

Na výstupu bude T řádků. Každý řádek odpovídá kontrole jednoho skřítků. Obsahem každého řádku je pak řetězec **anwa** nebo **cauca** v závislosti na tom, jestli skřítek překročil či nepřekročil zákonem povolenou dobu práce.

Ukázkové vstupy

Vstup

```
4
1
2 91
1
1 91
2
3 91
180 200
2
181 270
1 90
```

Výstup

```
cauca
anwa
anwa
cauca
```

Úloha č. 3

Květináč



Rozmysli, popiš a naprogramuj!

10 b

Blíží se Vánoce a po vzoru pozemských zvyků nařídil generál Stankevich vánočně vyzdobit hlavní náměstí. Součástí výzdoby je i květináč s vánočními stromečky. Každý stromeček je ozdobený daným počtem ozdob a potřebuje pravidelně dané množství vody. Vaším úkolem je vybrat, které stromečky do květináče zasadíte.

Hlavním požadavkem je pochopitelně připravit ten nejkrásnější květináč s tím, že máme pouze omezenou možnost zavlažování. Vzhled květináče je hodnocen pouze podle součtu ozdob na všech stromečcích v květináči. Počet stromečků v květináči není omezen, ale vzhledem k nutnosti je zavlažovat, musí platit, že množství vody přitékající do květináče musí stačit pro zavlažování stromečků v květináči.

Pro generálovu prezentaci máte nahlásit maximální možný celkový počet ozdob na stromcích ve vánočním květináči.

Vstup

Na prvním řádku je celé číslo N , $1 \leq N \leq 10^2$, značící počet stromků, ze kterých můžeme vybírat. Na druhém řádku je celé číslo F , $1 \leq F \leq 10^5$, značící maximální možný přítok vody ke květináči. Na dalších N řádcích jsou vždy 2 čísla p_i, s_i , $1 \leq p_i, s_i \leq 10^5$ značící pro každý stromek počet ozdob a spotřebu vody, v tomto pořadí.

Je zaručeno, že celkový součet ozdob na všech N stromcích i součet potřebných přítoků ke všem N stromkům je nejvýše 10^5 , neboli $\sum_{i=1}^N p_i \leq 10^5$ a $\sum_{i=1}^N s_i \leq 10^5$.

Výstup

Na výstupu bude jedno číslo P , značící maximální možný součet ozdob na stromcích v květináči tak, aby všechny stromky v květináči mohly být zavlažovány.

Ukázkové vstupy

Vstup

6
20
12 4
8 6
3 6
3 3
15 15
20 17

Výstup

27

Vstup

4
33
23 7
6 13
15 5
7 6

Výstup

51

Úloha č. 4

Půlnoční mše



Rozmysli, popiš a naprogramuj!

10 b

Na Marsu se, stejně jako na Zemi, slouží o Vánocích půlnoční mše. Jelikož se na Marsu nejedná o klasické pozemské Vánoce a fakticky se ani o mše nejedná, nazývají Martané tyto události jako námi nečitelnou směs znaků ve tvaru mše. Posádka ze Země, která na Marsu v tomto období přebývá, by však ráda oslavila Vánoce plnohodnotně. Všichni členové výpravy se tak shodli na tom, že se jedné z těchto mší zúčastní.

Všechny mše na Marsu začínají ve stejný čas, každá z nich však dle lokální tradice může trvat různě dlouhou dobu. Posádka má k dispozici plánek, na kterém jsou vyznačené lokality, kde se mše konají, a také poloha základny, kterou posádka obývá. Mars je však (jak jste se dozvěděli v minulých kolech) poměrně nehostinná lokalita, a tak se po jeho povrchu lze pohybovat jen po bezpečných cestách. Ty spojují výše zmíněné lokality a dále také křižovatky. Přejít z jedné cesty na druhou

pak lze, opět z bezpečnostních důvodů, jen na křižovatkách či v lokalitách. Posádka má v plánu naštěstí tyto cesty i křižovatky zaznačený také.

Jelikož posádka získala plánek povrchu Marsu od místního překupníka, jsou v něm obsaženy i další zajímavé informace. Zaprvé plánek obsahuje čas v ms (neboli v marsosekundách), udávající dobu potřebnou k uražení vzdálenosti dané konkrétními cestami (tj. jak dlouho trvá přejít z jednoho bodu (lokality/křižovatky) po bezpečné cestě do druhého bodu spojenými touto cestou). Vzhledem k hornatosti Marsu se doba potřebná k uražení vzdálenosti mezi body A a B může lišit od doby potřebné k uražení cesty z bodu B do A. Druhá informace, naškrábaná na druhé straně plánu, říká, že žádná z martanských lokalit/křižovatek není přímo napojena na více než 10 cest. To však pro posádku není nic nového, jelikož je obecně známo, že Martané se kvůli svému 6D vnímání v trojrozměrném prostoru pohybují obtížně.

Brzy ráno po mši se naše posádka účastní důležité průzkumné mise a musí tak vstávat v (ne)martansky brzký čas. Z toho důvodu se chtějí zúčastnit takové mše, ze které se po jejím skončení dostanou zpátky na základnu co nejdříve. Dokážeš jim na základě informací z plánu pomoci vybrat mši, které se mají zúčastnit? Lze předpokládat, že ze všech lokalit, ve kterých se konají mše, se dá dostat zpět na základnu.

Vstup

Na prvním řádku budou uvedena tři celá čísla N , L a M , $2 \leq N \leq 10^6$, $1 \leq L < N$ a $1 \leq M \leq \min(N(N-1), 5N)$, kde N udává součet počtu lokalit (základna a lokality, kde se odehrávají mše) a počtu křižovatek, L udává počet lokalit, ve kterých se konají mše a M udává počet cest v plánu. Pro účely identifikace má každý bod (lokality/křižovatka) na plánu přiřazeno číslo. Základna posádky má přiřazeno číslo 1, lokality, ve kterých se konají mše jsou číslovány od 2 do $L+1$ a křižovatky jsou číslovány od $L+2$ do N .

Poté následuje L řádků, každý obsahující jedno číslo T , $1 \leq T \leq 10^3$. Číslo na i -tém z těchto řádků určuje dobu trvání mše v lokalitě $i+1$. Následuje M řádků obsahujících tři celá čísla A , B a V , $1 \leq A, B \leq N$, $A \neq B$ a $1 \leq V \leq 10^3$, kde každý z řádků popisuje jednosměrnou cestu vedoucí z bodu A do B , jejíž zdolání trvá V marsosekund.

Výstup

Výstupem je jediné číslo - identifikátor lokality, ve které se koná mše splňující zadání výše. Pokud danou podmínku splňuje více lokalit, můžeš vypsát kteroukoliv z nich.

Ukázkové vstupy

Vstup

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

Výstup

3

Vstup

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

Výstup

2

Úloha č. 5

Bůh Mrkve



Rozmysli, popiš a naprogramuj!

10 b

Blíží se večer a jeden z místních starců se dal do vyprávění. Na jazyk mu přišla prastará legenda, pojednávající o dobrodruhovi a kouzelné jeskyni. . .

Žil byl jeden slavný dobrodruh, známý též pod pseudonymem Monroe. Dlouhé roky putoval po různých kontinentech hledajíc všelikteré poklady. Jednoho slunečného dne objevil překrásnou jeskyni. Jak brzy zjistil, jeskyně byla kouzelná. Ačkoli se v ní neukrýval žádný poklad, člověk si mohl přijít na slušné bohatství pouze tím, že v jeskyni chvíli pobyl. Za každou hodinu, kterou v jeskyni strávil se jeho peníze znásobily. Monroe samozřejmě neváhal a v jeskyni zůstal dlouhé a dlouhé hodiny. Když si řekl, že má konečně dost, pokračoval ve svém putování po rozlehlých krajích. Uplynulo několik dní, když tu narazil na velké město. Zjistil, že ve městě je velké množství hladovících sirotků, a jelikož byl velmi dobrosrdečný, rozhodl se jim vypomoci. Za všechny peníze, které bylo možno utratit, nakoupil mrkev a sirotky nakrmil. Tímto

činem si vysloužil nové přízvisko - Bůh Mrkve. Na jeho počest se nyní před Vánoci slaví svátky Mrkvevzdání.

Těmito slovy stařec ukončil své vyprávění. Pro tebe, jakožto matematického nadšence, to tímto ovšem zdaleka neskončilo. V příběhu bylo až příliš neznámých. Kolik měl dobrodruh na začátku peněz? Jakým koeficientem je kouzelná jeskyně přenásobovala? Jak dlouho v ní pobyl? Kolik stály mrkve? A nakonec – kolik mu po nákupu mrkví zbylo peněz? A to bude i tvým úkolem.

Vstup

Vstupní data budou obsahovat několik řádek. Na každé řádce budou čtyři čísla: **H**, **A**, **L** a **M**, které udávají počet peněz, kterým byl dobrodruh na začátku vybaven, koeficient přenásobení, počet dní, které strávil v jeskyni, a cenu jedné mrkve.

- $1 < H \leq 10^{18}$
- $1 < A \leq 10^{18}$
- $0 \leq L \leq 10^{18}$
- $1 \leq M \leq 2 \cdot 10^9$

Výstup

Na výstupu bude za každou řádku právě jedno číslo udávající, kolik peněz dobrodruhu po nákupu mrkví zbylo.

Ukázkové vstupy

Vstup

```
1 1 0 2
2 1 1 3
6 2 2 5
2 4 4 9
5 11 5 17
8 20 12 33
14 34 30 65
4 536870912 536870912 1073741825
```

Výstup

```
1
2
4
8
16
32
64
1073741824
```