

fiks!



ČESKÉ  
VYSOKÉ  
UČENÍ  
TECHNICKÉ  
V PRAZE

**FIT**

# Fitácký Informatický Korespondenční Seminář

Ročník 2015/2016, 4. 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 prozkoumání Marsu a oslavě Vánoc přišlo klidné období. Vše se na Marsu zaběhlo a nastal poklidný život, tak jako býval kdysi na Zemi. Nic však netrvá věčně a brzy přišel čas uvažovat, co dál. Mars je náš, ale vesmír je přece obrovský, tak proč se zastavit na Marsu? Bádání nikdy nekončí, tak neváhejme a pustme se do hlubin vesmíru! A co ty? Zůstaneš na Marsu, nebo pomůžeš s další výpravou a budeš s námi prozkoumávat svět algoritmů?

Volba je na tobě, buď tuhle brožurku odložíš a na FIKS zapomeš, nebo všem ukážeš, že na to máš. Stejně tak jako v předchozích kolech najdeš i v závěrečném 4. kole FIKSu několik vypečených úloh, nad kterými můžeš trápit svoje mozkové závity. Pokud jsi neřešil předchozí kola, neváhej a puť se do tohoto! Navzdory tomu, že někteří účastníci mohou mít bodový náskok z předchozích kol, stále 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 puť se do řešení úloh, které jsme pro tebe připravili!

*Tvoji organizátoři*

---

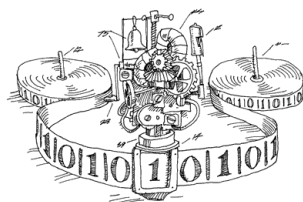
## Fitácký Informatický Korespondenční Seminář Ročník 2015/2016, 4. kolo

Začátek kola: 3. 2. 2016  
**Termín odevzdání: 13. 3. 2016**  
Odevzdávání: Přes webové rozhraní na <http://fiks.fit.cvut.cz>  
Další informace: <http://fiks.fit.cvut.cz>  
[kontakt@fiks.fit.cvut.cz](mailto:kontakt@fiks.fit.cvut.cz)

# fiks!

# Úloha č. 0a

## Lexikální analyzátor #1



---

Odpověz Sfinze!

1/10 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 10 %**.*

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

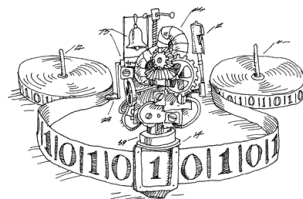
```
FALSE
TRUE
FALSE
```

---

---

# Úloha č. 0b

## Lexikální analyzátor #2



---

Odpověz Sfinze!

2.5/10 b

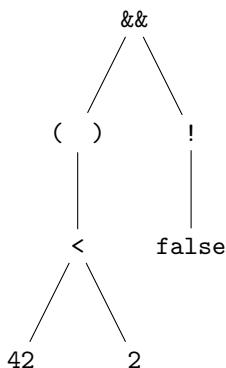
*Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.*

*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 2. kole, tzn. že její **maximální zisk je hodnocen pouze 25 %**.*

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 potřebuji 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

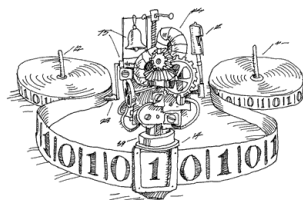
Výstup obsahuje  $N$  řádků. Každý řádek odpovídá jednomu zadání. Obsahem každého řádku je řetězec `true` nebo `false` charakterizující logickou hodnotu, která odpovídá vyhodnocení výrazu.

## Ukázkové vstupy

Vstup	Výstup
<code>true</code>	<code>true</code>
<code>0</code>	<code>false</code>
<code>27 &lt; 2</code>	<code>false</code>
<code>42    (false)</code>	<code>true</code>
<code>!false &amp;&amp; (23 &gt;= 3)</code>	<code>true</code>
<code>(true != true)    ((30 &gt; 2) &amp;&amp; !(29 &lt;= 1))</code>	<code>true</code>
<code>true != (!(!!true))</code>	<code>false</code>

# Úloha č. 0c

## Lexikální analyzátor #3



Odpověz Sfinze!

5/10 b

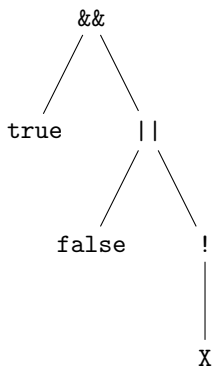
Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

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 3. kole, tzn. že její **maximální zisk je hodnocen pouze 50 %**.

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 Flegintonu vůbec nic neděje. Praporčík se rozhodl, že využije volný čas na rozšíření

jeho minulého programu. Občas si totiž lidé nejsou při vyhodnocování výrazů jisti 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 známe 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 0.1.



**Obrázek 0.1** *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 uzavorkovaný logický součet,
- řádně uzavorkovaný logický součin,
- název proměnné.

Uzavorkovaný 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

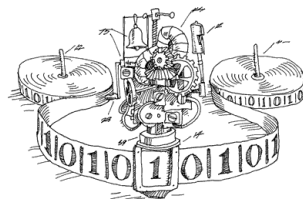
```
true
(true || false)
(!false && X)
(true || (X && false))
(!(true || X))
```

## Výstup

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

## Úloha č. 1

### Lexikální analyzátor #4



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.

A je to tady. Z Paxtona je podporučík. Je to hlavně důsledek jeho práce na vyhodnocovací výrazů. Podporučík své povýšení ale zatím slavit nemůže. Má zase plné ruce práce. Ukázalo se totiž, že jeho omezení na pouze jeden výskyt proměnné je značně omezující. Paxtonův první úkol v roli podporučíka tedy je toto omezení odstranit.

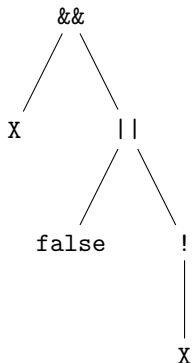
Na vstupu jsou *lexémy*, ze kterých je možné postavit syntaktický strom. Máme dva binární operátory, jeden operátor unární a v listech stromu se bude nacházet buď konkrétní hodnota, nebo zástupná proměnná. Závorky jsou jen pomocné a určují správnou prioritu operátorů. Příklad syntaktického stromu nalezneme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

## 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 uzavorkovaný logický součet,



**Obrázek 1.1** Syntaktický strom pro výraz  $(X \ \&\& \ (false \ || \ !X))$

- řádně uzavorkovaný 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. Proměnná může být v celém výrazu použita několikrát, ale vždy se jedná o stejnou proměnnou (stejně písmeno).

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

```

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

### Výstup

```

X
true
false
X
!X
  
```

## Úloha č. 2

### Taranský poklad



Rozmysli, popiš a naprogramuj!

10 b

„Kapitáne, zachytili jsme šifrované vysílání taranské flotily,“ zahlásil radista Kramer, „většinu zprávy jsme schopni rozluštit, ale část využívá nějaké neznámé šifrování!“ „Ukažte mi to,“ přikázal kapitán Steinbach.

*Poklad nezměrného množství zlata, platiny a dalších cenných kovů jsme zanechali na planetě se standardní vesmírnou adresou \*\*\*\*\* kontrolní číslo  $N$ .*

„Ten poklad musíme získat za každou cenu! Dokážete získat souřadnice té planety?“ zeptal se Steinbach. „Bohužel ne, pane. Víme přesně, v jakém tvaru Tarané posílají tajné zprávy, ale tu nejdůležitější část nemůžeme rozluštit. Známe sice kontrolní číslo a víme, jak ho spočítat, ale to nám pomůže zjistit adresu jen přibližně. Takové kontrolní číslo může být stejné pro více planet.“ „Tak je navštívíme pěkně jednu po druhé! Ti hloupí tvorové si myslí, že jejich poklady jsou před námi v bezpečí, ale to se mýlí!“ rozčílil se kapitán, „pro nás není nic nemožné, to si zapamatujte! Pošlete tu zprávu do výpočetního oddělení, tam už si budou vědět rady. A do večera chci znát počet těch planet!“

Tarané používají pro určení pozice planety ve vesmíru takzvanou standardní vesmírnou adresu. Ta se skládá z identifikačního čísla galaxie  $IDG$  a identifikačního čísla planety  $IDP$ , kde první číslo je vždy  $IDG$ . Obě čísla jsou přirozená a každá uspořádaná dvojice  $IDG$  a  $IDP$  označuje právě jednu planetu. Například Země má standardní vesmírnou adresu 252 1357. Dále víme, že Tarané někdy za takovou adresu přidávají kontrolní číslo  $N$ , které lze spočítat jednoduše jako  $N = \frac{IDG \times IDP}{IDG + IDP}$ . To jim pomáhá odhalit případné chyby v adrese, které mohou vzniknout při přenosu. Tarané však neradi počítají s jinými, než celými čísly, a proto kontrolní číslo přikládají pouze tehdy, pokud je přirozené. Tvým úkolem je zjistit pro zadané  $N$ , kolik standardních vesmírných adres odpovídá tomuto kontrolnímu číslu.

### Vstup

Vstup se skládá z jediného přirozeného čísla  $N$ ,  $1 \leq N \leq 10^9$ , značícího kontrolní číslo přiložené ke zprávě.

### Výstup

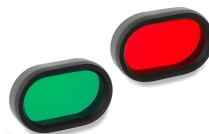
Na výstupu bude jedině číslo  $M$ , což je počet standardních vesmírných adres, kde se může nacházet poklad.

## Ukázkové vstupy

Vstup	Výstup
4	5
Vstup	Výstup
544	33
Vstup	Výstup
118918800	14175

## Úloha č. 3

### Filtry



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 průběhu novoroční oslavy se podnapilým průzkumníkům podařilo nečekaně objevit tajemnou síň plnou mimozemských záhad. Daná síň obsahuje mnoho komnat, kde každá je zabezpečena speciální matematickou hádankou. Před každou komnatou se nachází dva červeno-zelené obrazce a mezi nimi velké množství filtrů.

Obrazce i filtry jsou složeny z deseti částí. Na obrazcích je každá z částí buď červená, nebo zelená. Na filtrech jsou části propustné a nepropustné. Propustná část filtru barvu přirozeně propouští, zatímco nepropustná ji invertuje. Z červené se stane zelená, ze zelené červená. Cílem hádanky je aplikovat vybrané filtry tak, abychom pomocí nich změnili první obrazec na druhý.

Prolomení tohoto zabezpečení se vědcům dlouho nedařilo, až nakonec objevili další skryté zařízení. K prolomení je totiž potřeba do druhého zařízení zároveň vložit celkový počet správných kombinací filtrů. Kvůli omezeným schopnostem zařízení je navíc potřeba vložit počet kombinací modulo  $10^9 + 7$ .

### Vstup

Tato úloha má oproti tradičním úlohám drobně odlišný vstup. Kvůli velikosti vstupů je seznam filtrů volně dostupný na adrese <https://fiks.fit.cvut.cz/files/tasks/season2/round4/filtry.txt>. Každý řádek tohoto souboru obsahuje řetězec

dlouhý 10 znaků a představuje jeden filtr. Řetězce obsahují znaky 0 a 1, kde 0 představuje propustnou část a 1 nepropustnou část.

Dále obsahuje úloha již tradiční vstup. Na prvním řádku je číslo  $K$ , udávající počet tajemných komnat. Následuje  $K$  popisů komnat. Na prvním řádku popisu komnaty jsou dvě čísla  $B$  a  $N$ . Tato čísla popisují filtry dané komnaty. Tyto filtry se nachází ve výše zmíněném souboru s filtry a to na řádcích  $B+1$  až  $B+N$ . K dispozici tedy máme  $N$  filtrů. Další dva řádky obsahují desetiznakové řetězce skládající se ze znaků  $R$  a  $G$ , představující obrazce u dané komnaty.

Rozsah parametrů  $K$ ,  $B$  a  $N$  závisí na obtížnosti úlohy a to následovně:

- *Lehká varianta:*  $1 \leq K \leq 100, 0 \leq B \leq 10^3, 1 \leq N \leq 20$
- *Střední varianta:*  $1 \leq K \leq 100, 0 \leq B \leq 10^5, 1 \leq N \leq 10^3$
- *Těžká varianta:*  $1 \leq K \leq 200, 0 \leq B \leq 4 \cdot 10^6, 1 \leq N \leq 10^6$

## Výstup

Na výstupu bude  $K$  řádek. Na každém řádku bude odpověď pro jednu komnatu, tedy počet správných kombinací filtrů modulo  $10^9 + 7$ .

## Ukázkové vstupy

### Ukázka externího souboru

```
1010101010
1010000000
0000101010
1010000000
```

### Vstup

```
2
0 3
RGRGRGRGGG
GGGGGGGGRG
0 4
RRRRGGGRRR
GRRRGRGGR
```

### Výstup

```
2
4
```

## Úloha č. 4

### Vesmírný reportér



---

Rozmysli, popiš a naprogramuj!

10 b

Vesmírný reportér Egon se vydal na planetu Tatamta, aby zastihl velkou bitvu tří národů. Bobolové, Bolobové a Lobobové nedávno vyčerpali zdroje svých domovských planet, zatímco Tatamta je jediná planeta významná velikým nerostným bohatstvím široko daleko. Diplomacie je pro ně sprosté slovo, a tak nastupuje hrubá síla.

Když Egon přistál se svou vesmírnou lodí nad skalním převisem vedle bitevního pole, zjistil, že boj již skončil. Než se bude moci vrátit domů, musí zde udělat rozhovory s místními obyvateli a sestavit odhady padlých vojáků z jednotlivých národů. Takový odhad nelze jednoduše „střelit od oka“. Jeden špatný odhad by mohl poškodit jméno redakce v celém vesmíru. Proto musíme pečlivě spočítat, jaké ztráty armády utrpěly. Zde je situace komplikovaná tím, že poražený voják se teleportuje na náhodnou planetu. Na místě porážení zbude jeho trojúhelníkový amulet, který se rozpadne ve vrcholech. Víme, že každý Bobolan nosí amulet ve tvaru pravouhlého trojúhelníku, Bolobové nosí ostroúhlé a Lobobové tupouhlé. Egon si ve výpočtech nikdy moc nevěřil. Oslovil právě tebe s lákavou nabídkou, věnuje ti až 10 bodů, když to za něj vyřešíš. Dodá ti všechny tyčky, které posbíral. Tvým úkolem je zjistit, kolik maximálně mohlo padnout Bobolů, Bolobů a Lobobů.

### Vstup

Na prvním řádku je číslo  $M \in \langle 3, 1002 \rangle$ . Číslo  $M$  udává počet tyček. Následuje  $M$  řádek a na každé je právě jedno číslo udávající velikost tyčky  $S \in \langle 1, 32767 \rangle$ .

### Výstup

Výstupem jsou tři čísla oddělena mezerou, které popořadě reprezentují počet pravouhlých, ostroúhlých a tupouhlých trojúhelníků.

## Ukázkové vstupy

## Vstup

9  
3  
4  
5  
9  
9  
9  
5  
7  
10

## Výstup

2 31 28

## Úloha č. 5

### !SKIF

Rozmysli, popiš a naprogramuj!

10 b

Než budete moci odletět prozkoumávat širý vesmír, čeká vás ještě poslední úkol. Při odletu je nutné odposlouchávat komunikaci nepřátel, zda se nechystají podniknout proti vám nějaký útok. Jedním z podúkolů při této činnosti je zjišťovat četnost výskytů jednotlivých slov.

Tvým úkolem bude nalézt slovo, které se v odposlechu vyskytlo nejčastěji, a vypsát jeho četnost. Než však začneš s programováním, měl by sis přečíst zprávu, která ti přišla od náčelníka kmene, kterému jste při návštěvě Marsu velice pomohli.

zvarpa

zvidram sav pelatre! jmse vmlei rdai, ez jste man pmloohi. luimcoe es s vami. sav vtlieel im rkel o vaesm uoklu, a porto bcyh sav dar smezinal es sourrkuttu neasho jayzka. ej vntalse vlmei pbodnoy tomu vsmaeu, ej edz osvem nilkoek zidscaanh rlizdou. prvnm z nich ej, ez nase pmiso neni zekldaa kat bhaote, jako ot vase. nzmnaee yt vase hckay a craky dan rznuyimi psenmiy a zu vubec npzeiamuvoe zanda pmnesia vleka ic mala - ot yb dlealo pouze zkmtay. njeesti rzoidl vsak spvocia v mot, ez vesnhcy nsae nricae, ackoli uos jikkaolv oldisna, poavjuzi svloa az sjtena, pilazke es v nich vtyukyjsi snjeta piemsna - smzeomarje i ev stenejm pctou. dafuom ez toto bude scaitt k doeonkcní vhseao uloku. s paoezrdvm naeilnck knlau fakixsu tovmllala

## Vstup

Na prvním řádku bude počet provedených odposlechlů. Každý odposlechl začíná číslem  $N$  ( $N \leq 10^5$ ). Dále následuje  $N$  slov, která jsou od sebe oddělena jedním či více bílými znaky (tabelátor, mezera či konec řádky).

## Výstup

Za každý provedený odposlechl bude na výstupu řádek s právě jedním číslem, které značí největší četnost výskytu nějakého slova.

## Ukázkové vstupy

### Vstup

```
6
11
lla slatrom tsum liah ot regyt eht redael fo siht dnuor
6
there are three kinds of ether
4
dog is good god
9
roman moran called manro eating ramen in his manor
11
post lady wont stop smoking pots on her spot without tops
3
evangelist
is
evlisagent
```

### Výstup

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