Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb a přiděleného počítače (tzn. knihy, kalkulačky, mobily, apod.).

Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Řešení odevzdáváte pomocí soutěžního systému CMS, který ho automaticky otestuje na připravených sadách testovacích dat. Detaily hodnocení naleznete v letáku s popisem prostředí. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

Experimentální jazyky: V letošním ročníku MO-P je také možné odevzdávat praktické úlohy i v jazycích Python 3 a Java 11. U nich nicméně nezaručujeme, že bude možné získat plný počet bodů – je možné, že program nestihne doběhnout do časového limitu, byť používá algoritmus s optimální časovou složitostí.

P-III-4 Hyperjezdec

Jezdec je klasická šachová figurka. Pohybuje se tak, že ho zvedneme, posuneme jej o dvě políčka některým směrem (v řádku nebo sloupci), potom o jedno políčko tím druhým směrem a tam ho položíme.

Hyperjezdec se pohybuje podobně jako jezdec, jen rychleji: Nejprve jej posuneme nějakým směrem (v řádku nebo sloupci), potom ho posuneme **o jiný počet políček** v kolmém směru a tam ho položíme. V obou směrech se hyperjezdec musí pohnout, jeho nejkratší pohyby tedy odpovídají pohybům obyčejného jezdce.

Máme obří šachovnici s r řádky a s sloupci, na jejímž každém políčku je napsané jedno písmeno. Chceme po ní přeskákat hyperjezdcem tak, aby se postupně zastavil přesně na písmenkách daného slova w. Začít můžeme na libovolném políčku obsahujícím první písmeno w, každým dalším tahem se musíme přesunout na políčko obsahující následující písmeno slova w. Kolika různými způsoby to můžeme udělat? (Dva způsoby považujeme za různé, pokud jim odpovídají různé posloupnosti navštívených políček.)

Jelikož správná odpověď může být opravdu velká, počítejte pouze její zbytek po dělení číslem $10^9 + 7$.

Formát vstupu

Na prvním řádku dostanete dvě mezerou oddělená celá čísla r a s. Na druhém řádku dostanete slovo w. Na i-tém z následujících r řádků dostanete řetězec délky s popisující, která písmena jsou na i-tém řádku šachovnice. Můžete předpokládat, že všechna písmena na vstupu jsou velká písmena anglické abecedy.

Formát výstupu

Vypište řádek obsahující jedno celé číslo, totiž zbytek počtu různých způsobů, jak vyskákat na šachovnici slovo w, po dělení číslem $10^9 + 7$.

Příklady

Vstup:	*Výstup:*
3 4 HA HLOH NENI AKAT	2

Z písmena H vlevo nahoře nemůžeme skočit ani do jednoho z písmen A na spodním řádku. Naopak z písmena H vpravo nahoře lze skočit do obou písmen A na spodním řádku.

Vstup:	*Výstup:*
3 5 OREL VASEK HRAJE SACHY	0

Jelikož na šachovnici není žádné O ani L, je zřejmé, že se slovo OREL poskládat nedá.

Vstup:	*Výstup:*
3 5 UUUUUU UUKUU HESLO UUPUU	868

Z každého písmena U v rohu šachovnice lze skočit na tři jiná U. Z každého písmena U, které není v rohu, lze skočit jen na dvě jiná U: ta v protilehlých rozích.

Vstup:	*Výstup:*
6 8 KSP KSPPSKSP PPSKSKSP SKKSKSKP PPPSKSKK KKSKSKPP PPPSKSKP	1276

Jedno přípustné řešení je začít na K úplně vlevo nahoře, odtud se dostaneme na S zhruba uprostřed dole a odtud na P úplně vpravo nahoře

Bodování

V první sadě vstupů (1 bod) platí $r, s, |w| \leq 5$.

Ve druhé sadě vstupů (2 body) platí $r, s, |w| \leq 30$ a navíc je v každém vstupu dohromady nejvýše 10^5 různých způsobů, jak může hyperjezdec přeskákat nějaký prefix slova w.

Ve třetí sadě vstupů (2 body) platí $r, s, |w| \leq 40$.

Ve čtvrté sadě vstupů (2 body) platí $r, s, |w| \leq 100$.

V páté sadě vstupů (3 body) platí $r, s \leq 800$ a $|w| \leq 100$.

P-III-5 Kvádr

Michal vyhrál v tombole program, který vypisuje obsah trojrozměrného pole $pole[0 \ldots a-1][0 \ldots b-1][0 \ldots c-1]$ tak, že každé políčko vypíše právě jednou. Michalův program vypadá následovně:

```
for s = 0 to a+b+c-3:
    for i = 0 to a-1:
        for j = 0 to b-1:
            for k = 0 to c-1:
                if i+j+k == s:
                    print( pole[i][j][k] )
```

Soutěžní úloha

Na vstupu dostanete rozměry pole (čísla a, b, c) a několik otázek $n_1, \ldots, n_q$. Pro každou otázku n_i zjistěte souřadnice políčka, které Michalův program vypíše jako n_i-té v pořadí.

Formát vstupu

Na prvním řádku dostanete tři mezerou oddělená kladná celá čísla a, b, c. Na druhém řádku je kladné celé číslo q: počet otázek. Na třetím řádku jsou mezerou oddělená kladná celá čísla $n_1, \ldots, n_q$, jednotlivé otázky.

Formát výstupu

Pro každou otázku vypište jeden řádek obsahující tři mezerou oddělená celá čísla i, j, k: hledané souřadnice.

Příklady

Vstup:	*Výstup:*
3 3 3	0 0 0
3	0 0 1
1 2 10	2 0 0

Máme pole o rozměrech $3 \times 3 \times 3$. *Následují souřadnice prvních 10 políček, která program v tomto pořadí vypíše:* $(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(1,0,0)$, $(0,0,2)$, $(0,1,1)$, $(0,2,0)$, $(1,0,1)$, $(1,1,0)$ *a* $(2,0,0)$.

Vstup:

```
13 2 5
4
70 130 24 108
```

Výstup:

```
6 1 2
12 1 4
3 1 0
9 1 3
```

Bodování

Je deset sad vstupů, za každou je 1 bod. Ve všech sadách platí $q \leq 10^5$ a pro každé i je $1 \leq n_i \leq a \cdot b \cdot c \leq 10^{18}$. (Všimněte si, že na uložení hodnot a, b, c i n_i může být potřeba 64-bitová celočíselná proměnná.) V některých sadách platí různá další omezení dle následující tabulky:

číslo sady	$\max a$	$\max b, \max c$	$\max q$	další
1	20	20	20	
2	180	180	1	
3	2500	2500	1	
4	1	10^6		
5	1	10^9		
6	10^6		100	$a = b = c$
7	10^6			$a = b = c$
8	10^6	10^6	1	
9	10^6	10^6		
10				

P-III-6 Firma

Firma MixérLabs má strukturu úplného binárního stromu hloubky k. Každý zaměstnanec firmy má nějakou celočíselnou úroveň od 0 (stážisté) až po k (ředitelka celé firmy). Ředitelka je právě jedna a nemá žádného nadřízeného, každý jiný zaměstnanec má právě jednoho přímého nadřízeného. Stážisté nemají žádné podřízené, každý jiný zaměstnanec má právě dva přímé podřízené.

Po firmě si lidé posílají zprávy. Ty putují vždy jen mezi přímými nadřízenými a podřízenými, a to nejkratší možnou cestou. Jinými slovy, vždy, když má zaměstnanec x zprávu určenou pro zaměstnance y, postupuje následovně: Pokud je y (ne nutně přímým) podřízeným x, přepošle x zprávu tomu svému *přímému* podřízenému, v jehož oddělení y pracuje. V opačném případě přepošle x zprávu svému *přímému* nadřízenému.

Řekneme, že zaměstnanec b je v hierarchii *mezi* zaměstnanci a a c, pokud zpráva od a určená pro c projde přes b.

Zaměstnanci ve firmě mají svá navzájem různá identifikační čísla od 1 po $n = 2^{k+1} - 1$. (Pro každé číslo v tomto intervalu existuje právě jeden zaměstnanec, který jej má.) Chtěli bychom o celé firmě zjistit, kdo má jaké číslo.

Adam zná celou firmu, je ale ochotný odpovídat nám pouze na otázky o tom, jak po firmě chodí zprávy. Umíte pomocí nich zrekonstruovat celou hierarchii firmy?

Interakce

Vaší úlohou je napsat program, který bude komunikovat s „Adamem“ (naším programem). Váš program bude střídavě číst údaje ze standardního vstupu a psát je na standardní výstup.

Na začátku programu přečtěte ze vstupu dvě celá čísla k a ℓ: hloubku firemní hierarchie a limit na počet otázek, které je Adam ochotný zodpovědět.

Následně můžete klást otázky. Každou otázku položíte tak, že vypíšete řádek tvaru `0` a b c a následně přečtete Adamovu odpověď: Jedno číslo, které bude mít hodnotu 1, pokud je b v hierarchii mezi a a c, a 0, pokud není.

Pozor na to, že po každé otázce je potřeba vyprázdnit buffer standardního výstupu, aby se vypsaná data ihned dostala k Adamovi:

- V C nebo C++ se `stdio` pomocí `fflush(stdout);`
- V C++ s `iostream`y pomocí `cout << flush;`
- V Pascalu pomocí `flush(output);`
- V Pythonu stačí funkci `print` přidat argument `flush=True`.

Až si budete hierarchií jistí, váš program by měl vypsat $n + 1$ řádků a skončit. Na prvním z těchto řádků by mělo být číslo 1. Následně postupně pro každého zaměstnance v pořadí od čísla 1 do n vypište jeden řádek, na němž bude nejprve počet jeho přímých podřízených a potom jejich identifikátory (v libovolném pořadí, oddělené mezerami).

Ukázkový testovač

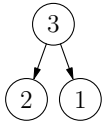
V adresáři `/mo/tasks/firma/` bude ukázková implementace „Adama“, kterou můžete použít při testování svých programů. Tento program nejprve přečte ze souboru `firma.in` popis firmy a potom výše uvedeným způsobem komunikuje pomocí svého standardního vstupu a výstupu.

Soubor `firma.in` má obsahovat dva řádky. Na prvním jsou dvě mezerou oddělená čísla k a ℓ. Na druhém je $n = 2^{k+1} - 1$ mezerou oddělených čísel: postupně pro každého zaměstnance v pořadí od 1 do n číslo jeho přímého nadřízeného, respektive 0, pokud jde o ředitelku.

Nyní popíšeme dva příklady, jak může komunikace programů vypadat. Soubory `firma.in` pro oba příklady najdete v adresáři `/mo/tasks/firma/`.

Příklady

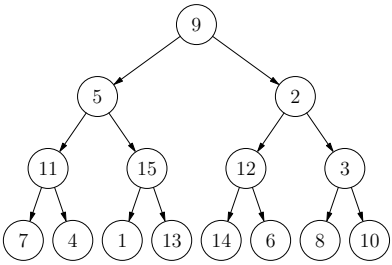
Na obrázku je struktura firmy s $k = 1$:



Následuje příklad kompletní komunikace Adama s vaším programem. Čtěte ho shora dolů. Prázdné řádky tam jsou pouze pro přehlednost, ve skutečné komunikaci se neobjeví.

Adam	*váš program*	*komentář*
`1 250000`		$k = 1$, $\ell = 250\,000$
	`0 1 2 3`	Leží 2 mezi 1 a 3?
`0`		Ne.
	`0 1 3 2`	Leží 3 mezi 1 a 2?
`1`		Ano.
	`1`	Už znám řešení.
	`0`	Zaměstnanec 1 nemá žádné podřízené.
	`0`	Zaměstnanec 2 nemá žádné podřízené.
	`2 1 2`	Zaměstnanec 3 má dva přímé podřízené.

Na dalším obrázku je příklad větší hierarchie firmy s $k = 3$:



Pro tuto firmu by odpověď „ano“ (`1`) dostaly například otázky `4 5 9`, `3 2 1` či `12 12 6`. Odpověď „ne“ (`0`) by dostaly například otázky `7 13 2`, `12 11 6` či `3 9 8`.

Bodování

Program, s nímž budete interagovat („Adam“) se nesnaží být zákeřný. V každém testu je celá hierarchie firmy předem zvolená a Adam vám opravdu odpovídá jen na otázky o ní. Můžete předpokládat, že na každou otázku umí odpovědět v zanedbatelném čase.

Ve všech sadách platí $k \geq 1$. V každé sadě mají všechny testy tutéž hodnotu ℓ. Následuje tabulka s popisem jednotlivých sad:

číslo sady	*body*	$\max k$	ℓ
1	3	5	250 000
2	1	5	50 000
3	2	6	100 000
4	3	10	800 000
5	1	11	500 000