



## Úlohy 2. kola zimnej časti

Termín odoslania riešení tejto série je pondelok 5. januára 2015.

### 1. Zwarte doos 2

kat. Z; 0 b za popis, 10 b za program

Zwarte doos mala v KSP obrovský úspech. Každý jeden vedúci si ju chcel vyskúšať a prejsť. Lepší suvenír z Holandska Sysel doniesť ani nemohol.

Vedúci sú však šikovní a po pár dňoch mal už každý všetkých 10 problémov vyriešených. A doos zapadla prachom kdesi hlboko v skrini. Nebola tam však sama. Okrem nej tam totiž ležali rôzne spoločenské hry, klávesnice, nefunkčné LAN káble a kopa knižiek. Väčšinou matematických, ale našli by ste tam aj zopár slovenských klasikov.

Keď sme minulý týždeň upratovali, všimli sme si, že doos dlhé chvíle efektívne využila, načerpala inšpiráciu a vymyslela si nových 10 úloh. Bola by škoda sa s vami o ne nepodeliť.

#### Úloha

Dostanete prístup k simulátoru čiernej skrinky. Vždy, keď do nej vložíte nejaké číslo, skrinka niečo vypíše (číslo, slovo...) podľa jednoduchého pravidla.

Vašou úlohou bude toto pravidlo odhaliť a nájsť vstupné číslo také, aby skrinka vypísala požadovanú vec.

Hra má 10 úrovní, za každú môžete získať bod. Náročnosť by mala stúpať spolu s úrovniami, môžete ich však riešiť v ľubovoľnom poradí.

#### Odovzdávanie a bodovanie

Za každú vyriešnú úroveň dostanete jeden bod. K tejto úlohe netreba odovzdávať žiadny popis ani program. Simulátor nájdete na stránke <http://ksp.sk/specialne/ksp/32/2/1>. Pre prístup na stránku musíte byť prihlásení a body sa vám budú pripočítavať automaticky.

### 2. Zobudená programátorka

kat. Z; 6 b za popis, 4 b za program

Kamila má rada kávu. Rokmi pokusov zistila presný objem kávy, pri ktorom sa jej pracuje najlepšie. Dnes sa dozvedela, že má náročnú domácu úlohu, ktorú treba rýchlo odovzdať, a potrebuje sa posilniť. Zamierila teda ku svojej zbierke výberových káv. Kávy v zbierke sú uskladnené v rôzne veľkých vrecúškach: každé vrecúško je dvakrát väčšie ako to pred ním, počínajúc vrecúškom s hmotnosťou 1 gram. To znamená, že v zbierke má vrecúška s váhami 1, 2, 4, 8, ...,  $2^n$  gramov. Ako v každej správnej zbierke, aj v tej Kamilinej je vrecúško každej veľkosti práve raz.

Ktoré vrecúčka má Kamila vybrať aby dosiahla svoju vytúženú hladinu kofeínu?

#### Úloha

Na vstupe dostanete  $n$  – potrebnú hmotnosť kávy a máte zistiť, ktoré z vrecúšok má Kamila zobrať, ak chce mať presne  $n$  gramov kávy.

#### Formát vstupu

Vstup má jeden riadok a v ňom jedno celé číslo  $n$ . Môžete predpokladať, že  $1 \leq n \leq 10^{18}$ .

#### Upozornenie

Dajte si pozor na to, že váš program musí pracovať aj s hodnotami, ktoré sa **nezmestia** do bežnej (32-bitovej) celočíselnej premennej. Na ich uloženie potrebujete použiť premennú s dostatočne veľkým rozsahom – napríklad `int64` v Pascale alebo `long long` v C++.

Takisto si dajte pozor, že súčin dvoch 32-bitových premenných vráti 32-bitové číslo bez ohľadu na to, či sa tam výsledok zmestí. T.j. `x := 1000000 * 1000000` uloží do `x` hodnotu 1420103680 bez ohľadu na to, či `x` je `int64` alebo `longint`.

## Formát výstupu

Vypíšte jediný riadok a v ňom použité vrecúška oddelené medzerou v poradí od najmenšieho po najväčšie. Nezabudnite ukončiť riadok znakom konca riadku. Teda napríklad v Pasmale vypíšte výsledok volaním `writeln(vysledok)`, v C++ zase volaním `cout << vysledok << endl`.

## Príklad

vstup	výstup
20	4 16
vstup	výstup
35	1 2 32

## 3. Zved Tigrík

kat. Z; 6 b za popis, 4 b za program

Každý z vás určite pozná "Súkromné očká". Podstatnou náplňou ich práce je zháňať informácie. Rozlišujú práve 26 typov informácií a každá z nich má kódove označenie podľa jedného veľkého písmenka abecedy.

V tejto organizácii pracuje aj Tigrík (jeho pravé meno je, samozrejme, iné). Jeho úlohou je zháňať informácie rôznych typov a doniesť ich svojmu šéfovi. Je platený za každý deň samostatne.

Tigrík má práve  $n$  tajných zdrojov. Každý z nich má informáciu jedného typu. Za jeden deň vie obehať iba  $k$  svojich zdrojov (potrebuje ich stretnúť osobne, inak by mu nič nepovedali).

"Súkromné očká" sú za svoj deň hodnotení nasledovne: ich šéf si vezme zoznam informácií, ktoré za daný deň doniesol konkrétny zamestnanec a každej priradí spoľahlivosť. Spoľahlivosť jednej informácie je rovná počtu informácií toho istého typu, ktoré dané očko donieslo. Plat, ktorý pracovník za deň dostane, je celkový súčet spoľahlivostí všetkých informácií. Napríklad za prinesenie 3 informácií typu A a 2 informácie F dostane  $3 + 3 + 3 + 2 + 2 = 13$  peňazí.

Tigríka by zaujímalo, koľko najviac peňazí môže za konkrétny deň dostať.

### Úloha

Na vstupe dostanete počet Tigríkových tajných zdrojov a počet zdrojov, ktorý dokáže Tigrík za daný deň obehať. Okrem toho dostanete reťazec  $n$  znakov, v ktorom  $i$ -ty znak predstavuje typ informácie  $i$ -teho zdroja.

Zistite, koľko najviac môže Tigrík za tento deň zarobiť. Nezabudnite správnosť svojho riešenia odôvodniť.

### Vstup

Prvý riadok vstupu obsahuje prirodzené čísla  $n$  a  $k$  ( $1 \leq k \leq n \leq 1\,000\,000$ ).

Druhý riadok obsahuje  $n$  veľkých písmen anglickej abecedy – typy informácií, ktoré majú tajné zdroje.

### Výstup

Vypíšte jeden riadok a v ňom jedno číslo – najväčší súčet spoľahlivostí, aký vie Tigrík dosiahnuť.

### Príklady

vstup	výstup
15 10 DZDFDFZDFDDDDDDF	82
vstup	výstup
6 4 YJSNPI	4

V prvom príklade prinesie 9 krát D a raz Z. V druhom príklade prinesie informácie 4 rôznych typov.

## 4. Záhradka

kat. Z a O; 6 b za popis, 4 b za program

Kleofáša omrzelo kšeftovať s vysávačmi a rozhodol sa dať sa na farmárčinu. Krok jedna je samozrejme zohnať si záhradku, a na takú záhradku treba plot. Preto sa Kleofáš rovno vybral kúpiť tyčky a pletivo. A keďže mali skvelú zľavu na  $s$  tyčiek, rovno ich kúpil. Následne si kúpil aj kozu, o ktorej vie, že potrebuje  $t$  metrov štvorcových trávy. A teraz rozmýšľa, ako má vyzeráť jeho záhradka, ak chce byť ekonomický – nenechať žiadne

tyčky vyjsť navnivoč (t.j záhradka má mať práve  $s$  rohov, teda aj  $s$  strán) a dať koze práve  $t$  m<sup>2</sup> trávy. Zjavne ekonomickým metariešením je kúpiť si na vyriešenie tohto problému milého študenta ...

### Úloha

Dané sú  $s$  a  $t$  také, že  $t \geq s/2$ . Kleofášova záhradka sa má dať nakresliť do štvorcovej siete (t.j. dĺžky strán musia byť celočíselné a všetky priľahlé kusu plotu musia byť na seba kolmé) a musí byť súvislá. Nájdite takú záhradku, ktorá má obsah  $t$  a  $s$  strán, alebo vypíšte "Neda sa", ak sa to nedá.

Navyše v popise riešenia dokážte, že existuje nekonečne veľa dvojíc  $(s, t)$ , pre ktoré  $t < s/2$  a napriek tomu takáto záhradka existuje.<sup>1</sup>

### Vstup

V jedinom riadku vstupu sú dve čísla:  $s$  a  $t$ , pričom  $4 \leq s \leq 10^3$  a  $1 \leq t \leq 10^6$ . Vždy platí, že  $t \geq s/2$ .

### Výstup

Vypíšte  $s$  riadkov: návod, ako má Kleofáš postaviť plot. Každý riadok je vo formáte  $S D$ , kde  $S$  je smer – jedno z písmen "S", "J", "V" a "Z" podľa svetovej strany a  $D$  je dĺžka tohto kusu plotu.

Záhradka musí byť uzavretá (teda musí skončiť tam, kde začala) a nasledujúce strany musia byť na seba kolmé.

Niektoré vstupy budú malé ( $s, t < 25$ ). Pre niektoré vstupy platí  $t \geq s$ . Nejaké body teda získate aj za čiastkové riešenia.

### Príklad

vstup	výstup
6 5	V 1 S 1 V 1 J 3 Z 2 S 2
vstup	výstup
9 12	Neda sa

## 5. Odpad spoločnosti

kat. Z a O; 7 b za popis, 8 b za program

Obaly od orechovníka, októbrové olovranty, omrvinky od obilných otrúb, odvädnuté obzerance, oreganom ochutená omáčka...

Hrôza. To všetko nájde pani upratovačka v školských laviciach. Každý deň. V jednej triede. Je to na nevydržanie. Ak náhodou stretne nejakého odchádzajúceho študenta a spýta sa ho, čie veci sú v lavici, odpoveď je vždy rovnaká: "Spolusediaceho." A Janko je fuč. Hlavne, že všetci vychvaľujú, ako čisto majú vo Viedni... a doma si takýto bordel nechajú...

Keďže anonymní spolusediaci sa ťažko obviňujú, rozhodla sa pani upratovačka, že si pekne zistí, kto s kým sedí. Ak náhodou s Jankom sedí Miško, nabudúce mu to poriadne vytmaví. Cez vyučovanie do triedy vojsť nemôže, takže musí vymyslieť iný spôsob ako nájsť dvojice obyvateľov zaprataných lavíc.

Našťastie si pani upratovačka posledných desať rokov zapisuje, koľko lavíc musela vypratávať každý deň. Ak navyše použije záznamy o dochádzke z triednej knihy, nič ju nemôže zastaviť. "Ha ha ha." Jedine mesiac listovania v zoznamoch... ale na to ste tu vy, dobrosrdeční riešitelia.

### Úloha

Táto úloha je interaktívna. Namiesto kompletného vstupu budete dostávať odpovede na vaše otázky.

Pre každú testovaciu sadu dostanete najprv číslo  $n$ , udávajúce počet lavíc v triede. Do triedy chodí  $2n$  študentov, číslujeme ich od 1 po  $2n$  a v každej lavici sedia práve dvaja. Vašou úlohou je zistiť ktoré dvojice študentov sedia spolu.

Zdrojom vašich informácií sú záznamy o dochádzke a zápisník pani upratovačky. Váš program sa môže spýtať na ľubovoľnú množinu žiakov a zistiť v kolkých laviciach dokopy sedia. (Teda napríklad ak 1 sedí spolu s 2, tak  $\{1, 2\}$  sedia v jednej lavici,  $\{1, 3\}$  sedia v dvoch laviciach a  $\{1, 2, 3\}$  sedia tiež v dvoch laviciach.)

<sup>1</sup>T.j. nájdite konštrukciu, ktorá funguje pre nekonečne veľa takých dvojíc.

Môžete sa však spýtať **najviac 20 000 otázok**.

### Formát vstupu

Na prvom riadku vstupu je malé kladné celé číslo  $t$  udávajúce počet sád.

Pre každú sadu dostanete na novom riadku kladné celé číslo  $n$  neprevyšujúce 1 000, určujúce počet lavíc v triede.

Následne testovač odpovedá na každú vašu otázku vypísaním jedného celého čísla na nový riadok – počet lavíc, v ktorých títo žiaci sedia. Túto odpoveď viete načítať zo štandardného vstupu.

### Formát výstupu

Na počet lavíc, v ktorých sedí  $m$  študentov sa spýtate vypísaním čísla  $m$  a následne vypíšete čísla študentov  $s_1, s_2, \dots, s_m$ , o ktorých sa zaujímate do jedného riadku oddelených medzerami. Aby bola vaša otázka spracovaná **potrebujete** výstup presunúť z pamäte na štandardný výstup pomocou príkazu `fflush(stdout)`; v C++, alebo `flush(output)`; v Pascale.

Keď ste si istí, že poznáte všetky dvojice, vypíšete 0 na nový riadok a  $n$  riadkov dvojíc čísel študentov, ktorí sedia spolu.

Pokiaľ sa spýtate viac ako 20 000 otázok, váš program bude nemilostrdne ukončený, avšak v testovači sa to môže prejavíť rôzne. Odpovede “prekročený časový limit”, “chyba počas vykonávania programu” aj “zlá odpoveď” môžu ale nemusia v skutočnosti zmanenaf, že ste sa opýtali priveľa otázok.

### Príklad

vstup	výstup
<pre>2 // počet sád 3 // počet lavíc v prvej sade 2 1 2 2 2 // počet lavíc v druhej sade 1</pre>	<pre>3 1 2 3 2 1 2 2 3 4 2 3 5 0 1 2 3 6 4 5 2 1 2 0 1 2 4 3</pre>

Skutočné vstupy neobsahujú “//” ani text za týmito znakmi. Pridali sme ich do ukážkového vstupu kvôli prehľadnosti.

Najprv sa dozvieme, že študenti 1,2,3 obsadzujú 2 lavice. Zistíme, že 1,2 sedia spolu. Treťou a štvrtou otázkou zistíme, že študent 3 nesedí so 4 ani s Petkou, teda musí sedieť so spolužiakom 6. Zvyšní dvaja študenti, 4,5 musia teda sedieť spolu.

V druhej sade sa spýtame na dvojicu 1,2. Keď sa dozvieme, že sedia spolu, už je zjavné, ako trieda vyzerá.

## 6. Opatera zvieratiek

kat. O; 10 b za popis, 10 b za program

Matfyzák Ondrej sa už v škole nudí, a tak chodí pomáhať do terárií blízkej zoologickej záhrady. Tam sa rád stará o hlodavce. Kími nimi písmenkovú veľužovku, ktorú má ešte radšej ako hlodavce. Pozdĺž celého tela má totiž škrvny tvaru písmen. A aj jej spôsob života sa Ondrovi veľmi pozdáva. Celý deň len vylihuje a žerie<sup>2</sup>. Keby vám však niekto hovoril, že je Ondro lenivý, tak klame. Napríklad teraz usilovne premýšľa, ako zlepšiť pochmúrnu zimnú náladu v teráriách.

Nakoniec sa rozhodol, že našej užovke vykúzlil úsmev na tvári pekným menom. Aby bolo osobné, chce vybrať nejakú podpostupnosť písmen na jej tele. Ondrej ale vyznáva princípy KSP – krása, symetria, pisateľnosť, preto je pre neho problém vybrať správne meno. Krásne meno je čo najdlhšie. Symetrické sa číta spredu rovnako ako zozadu – je to palindróm. A pisateľné meno sa dá napísať na cedulku pred teráriom. Experimentálne už zistil, že na cedulku nevopchá viac než sto znakov. Pomôžte mu vybrať nejaké KSP meno!

<sup>2</sup>Veľužovka, nie Ondro.

## Úloha

Podpostupnosť reťazca dostaneme vynechaním niektorých znakov. Podpalindróm je podpostupnosť znakov ktorá je palindrómom. Máme užovku popísanú ako reťazec písmen. Nájdite ľubovoľný podpalindróm dlhý 100 znakov, alebo najdlhší možný, ak tam vhodný stopísmenový nie je.

### Formát vstupu

Jediný riadok vstupu obsahuje reťazec do  $10^6$  malých písmen anglickej abecedy – popis veľžovky.

### Formát výstupu

Vypíšte reťazec  $m$  – vami odporúčané meno podľa princípov KSP. Ak existuje viacero rovnako dobrých mien, vypíšte ľubovoľné z nich.

### Príklady

vstup	výstup
abcdabcd	bab
vstup	výstup
napisaasipan	napisaasipan
vstup	výstup
abcdefghijklmnopqrstuvwxyz	f

## 7. Osobná doprava

kat. O; 8 b za popis, 12 b za program

Táto úloha má tak dlhé zadanie, že by potrebovala mínus pätnásť riadkov rozprávky.

Máme krajinu a v krajine rozmiestnených niekoľko zastávok. Zastávky majú mená: reťazce 1 až 10 malých písmen anglickej abecedy. Medzi  $d$  dvojicami zastávok sa dá priamo cestovať. Pre takéto dvojice zastávok máme zadanú vzdialenosť v metroch. (Vzdialenosť je rovnaká oboma smermi.)

V našej krajine existuje  $s$  jednosmerných spojení. Každé spojenie postupne navštívi dve alebo viac zastávok. Popis spojenia obsahuje okrem zoznamu zastávok ešte tri parametre: jeho rýchlosť  $v_i$  (v metroch za sekundu), jeho periódu  $p_i$  (v sekundách) a jeho offset  $o_i$  (tiež v sekundách). Čas potrebný na presun medzi dvoma zastávkami si vieme vypočítať tak, že vzdialenosť medzi nimi vydáme rýchlosťou a výsledok zaokrúhlime **nahor** na celé sekundy. Význam offsetu a periódy je nasledovný: Po trase spojenia už od nepamäti každých  $p_i$  sekúnd vyráža zo začiatkovej zastávky nový spoj. Najbližšie sa tak stane o  $o_i$  sekúnd odteraz.

Všimnite si, že pre jednoduchosť predpokladáme, že sa perióda nemení počas dňa. A taktiež sme zanedbali čas státia na zastávke: všetky naše spoje na zastávkach stoja 0 sekúnd, nastupuje a vystupuje sa za jazdy :)

Medzi spojeniami vieme na zastávkach ľubovoľne prestupovať. Na prestup nám tiež stačí nulový čas. Ak teda zastávkou prechádzajú v tom istom okamihu dva rôzne spoje, stíhame prestúpiť z jedného na druhý. Na zastávkach samozrejme môžeme na prestup aj ľubovoľne dlho čakať.

## Úloha

Daný je popis siete zastávok a spojení medzi nimi. Následne nasleduje niekoľko otázok. Každá otázka je tvorená dvomi menami zastávok: odkiaľ a kam chceme ísť. Vašou úlohou je vypočítať, či sa to dá, a ak áno, v akom najkratšom čase. Presnejšie, ak sme v čase 0 na zastávke, odkiaľ ideme, v akom najskoršom čase vieme byť na zastávke, kam ideme?

### Formát vstupu

V prvom riadku vstupu je číslo  $d$ , udávajúce počet dvojíc zastávok, medzi ktorými sa dá priamo cestovať. V každom z ďalších  $d$  riadkov vstupu je popis jednej dvojice zastávok. Tento je uvedený vo formáte “meno1 meno2 vzdialenosť”.

V nasledujúcom riadku je číslo  $s$ , udávajúce počet spojení. V každom z nasledujúcich  $s$  riadkov je popis jedného spojenia. Tento je uvedený vo formáte “ $v_i p_i o_i z_i$  zastavka\_1 . . . zastavka\_ $z_i$ ”. Významy  $v_i$ ,  $p_i$  a  $o_i$  sú uvedené vyššie, číslo  $z_i \geq 2$  udáva počet zastávok obsluhovaných spojením. Zastávky sú navzájom rôzne a pre každú dvojicu po sebe idúcich zastávok sme na vstupe mali uvedenú ich vzdialenosť.

V nasledujúcom riadku je číslo  $q \leq 10$ , udávajúce počet otázok. V každom z ďalších, posledných  $q$  riadkov vstupu je popis jednej otázky. Tento je uvedený vo formáte “odkial kam”, pričom odkial a kam sú platné mená dvoch rôznych zastávok.

### Obmedzenia

Vo všetkých vstupoch platí  $d \leq 300\,000$ , ale v približne polovici vstupov je  $d$  výrazne menšie. Rôznych zastávok bude nanajvýš 100 000.

Všetky vzdialenosti medzi zastávkami, rýchlosti a periódy sú **kladné** celé čísla neprevyšujúce 100 000. Pre offesity platí  $0 \leq o_i < p_i$ .

Súčet všetkých  $z_i$  (teda počtov zastávok jednotlivých spojení) neprekročí 300 000. Vo vstupoch s malým počtom zastávok bude aj počet spojení malý.

Optimálna cesta (ak nejaká existuje) vždy trvá menej ako 20 dní. Všetky potrebné výpočty by sa vám teda mali zmestiť do bežných celočíselných premenných.

### Formát výstupu

Pre každú otázku uveďte jeden riadok a v ňom text “neda sa” ak sa z daného začiatku do daného cieľa nedá dostať, resp. text “?d ?h ?m ?s”, kde namiesto otáznikov uveďte najmenší možný počet dní, hodín, minút a sekúnd, po ktorom vieme byť v cieľi cesty.

### Príklad

vstup	výstup
<pre>7 skladka smetisko 350 kontajner smetisko 299 dub javor 123 javor breza 234 dub breza 45678 breza lipa 1000 topol breza 50010 6 15 600 47 3 skladka smetisko kontajner 23 10 0 3 dub breza javor 1 1234 5 4 dub javor breza lipa 4 350 35 3 dub javor breza 100 1 0 2 javor dub 10 50 0 3 topol breza lipa 3 skladka kontajner kontajner skladka dub lipa</pre>	<pre>Od 0h 1m 31s neda sa Od 0h 4m 11s</pre>

Prvé spojenie má rýchlosť  $v_i = 15$  (bežná MHD), periódu  $p_i = 600$  (ide raz za 10 minút) a offset  $o_i = 47$ . Premáva na trase skladka – (350 m) – smetisko – (299 m) – kontajner. Vzdialenosť medzi skládkou a smetiskom prekonajú spoje tejto linky za 24 sekúnd, vzdialenosť medzi smetiskom a kontajnerom za 20. Najbližšie tri spoje po tejto trati odídu od skládky o 47, 647 a 1247 sekúnd odteraz, prejdú okolo smetiska o 71, 671 a 1271 sekúnd odteraz, a svoju cestu ukončia pri kontajneri o 91, 691 a 1291 sekúnd odteraz. Ak pôjdeme prvým z nich, dostaneme sa teda ku kontajneru o 1 minútu a 31 sekúnd od začiatku.

Pozrime sa teraz na poslednú otázku: ako sa dostať od dubu k lipe?

Ako prvé odchádza už v čase 0 spojenie, ktoré rýchlosťou 23 smeruje k breze a odtiaľ k javoru. Síce by sme sa ním mohli odviezť k breze, to ale nie je veľmi dobrý nápad – prišli by sme tam až po 1986 sekundách, a ešte by sme sa museli odtiaľ dostať k lipe.

Ide nám aj priama linka okolo javora a brezy k lipe, ani tou však nie je dobrý nápad ísť. Najbližší spoj síce ide už o 5 sekúnd, je však pomalý: potrvá mu to 123 sekúnd k javoru, ďalších 234 ku breze a ďalších 1000 k lipe. Do cieľa by sme teda dorazili až po 1362 sekundách.

Najlepšie riešenie vyzerá nasledovne: Na štarte počkám 35 sekúnd, potom nasadnem na spoj idúci rýchlosťou 4 okolo javora k breze. Ten ma za 31 sekúnd privezie k javoru a za ďalších 59 k breze. Tam vystúpim. Je práve 125 sekúnd od začiatku. O ďalších 26 sekúnd, teda v čase 151, pôjde okolo brezy spoj na linke topol-breza-lipa.

(Všimnite si, že tento spoj vyrazil na svoju cestu výrazne skôr ako my.) Ten ma za 100 sekúnd prevezie od brezy k lipe. V cieľi som teda po 251 sekundách.

## 8. Okná sa vymieňajú

kat. O; 15 b za popis, 10 b za program

Na matfyzе sa v rámci prestavby začali vymieňať všetky okná. A to zahŕňa aj KSPácku miestnosť T2. A aby sme uvoľnili miesto robotníkom, musíme ju celú vypratať. Problém však je, že v T2 sa nachádza **všetko**. A tým myslím naozaj všetko. Od gitary, počítačov, hlavnovedúcovskej tyče, soba, masky Darth Vadera, cez chemikálie, terčovnicu, vrtáčku, papuče, po pílku na železo, hasiaci prístroj, TODO list a nič.<sup>3</sup> Navyiac má každá z týchto vecí iný objem a keďže sa tu nachádza všetko, pre každé prirodzené číslo existuje v T2 jeden predmet s takýmto objemom.

Našťastie ako informatici máme dobrý komprimovací prístroj, ktorý dokáže bezstratovo zmenšiť objem ľubovoľného objektu s objemom väčším ako 1. Tento komprimátor funguje tak, že ak mal predmet pôvodne objem  $a$ , po skomprimovaní bude mať tento predmet objem  $b$ , kde  $b$  je počet jednotiek v binárnom zápise čísla  $a$ . Ak sa napríklad  $a = 19_{10} = 10011_2$ , tak  $b = 3$ . Teraz je snáď jasné, prečo sa predmet s objemom 1 nedá komprimovať. Táto komprimácia sa dá samozrejme opakovať, čím dostávame stále menšie a menšie predmety, až kým sa dostaneme na objem 1. Pre  $a = 19$  musíme komprimáciu opakovať 3 krát, pričom dostanem postupne objemy 3, 2 a 1.

Avšak skôr ako sa komprimátor začal používať, nadšení a akciechtíví prváci vypratali časť T2 a zostali v nej len predmety, ktorých objemy sú medzi  $l$  a  $h$  vrátane. Všetky tieto predmety chceme teraz skomprimovať na objem 1, musíme si však dať pozor, aby sme ich vedeli dekomprimovať. Dôležité je, aby sme každý predmet dekomprimovali presne toľko krát, koľko krát sme ho komprimovali. Všetky veci (v tom čase už s objemom 1) si teda uložíme do krabíc tak, aby veci, ktoré potrebovali rovnaký počet komprimácií, kým skončili s objemom 1, boli v tej istej krabici. Dopredu by sme však chceli vedieť, aké veľké krabice si máme pripraviť na ktorú sadu predmetov.

### Úloha

Pre čísla  $l$ ,  $h$  a  $k$  určite počet takých predmetov s objemom medzi  $l$  a  $h$  vrátane, že na ich skomprimovanie na veľkosť 1 je potrebných práve  $k$  komprimácií.

### Formát vstupu

Na prvom riadku vstupu je číslo  $t$  ( $1 \leq t \leq 1000$ ) – počet vstupných testov.

Nasleduje  $t$  riadkov, každý obsahujúci tri čísla  $l$ ,  $h$  a  $k$  ( $2 \leq l \leq h \leq 10^{18}$ ,  $1 \leq k \leq 10^6$ ).<sup>4</sup>  $l$  a  $h$  udávajú interval objemov, z ktorého vyberáme a  $k$  je žiadaný počet komprimácií.

### Formát výstupu

Pre každý z  $t$  testov vypíšte jedno číslo – počet takých čísiel medzi  $l$  a  $h$ , že sa skomprimujú na 1 po práve  $k$  komprimáciách.

### Bodovanie

Vstupy sú rozdelené do 10 testovacích sád. Jednotlivé sady majú svoje obmedzenia a väčšinou platí, že vyriešiť skoršiu sadu je jednoduchšie. Zároveň by riešenia niektorých nižších sád mali pomáhať pri vymýšľaní zložitejších riešení.

- Pre sadu 01 platí, že  $t = 100$ ,  $b \leq 10^6$ ,  $b - a \leq 1000$
- Pre sadu 02 platí, že  $t = 100$ ,  $b \leq 10^{18}$ ,  $b - a \leq 1000$
- Pre sadu 03 a 04 platí, že  $t = 1000$ ,  $b \leq 10^{18}$ ,  $b - a \leq 10^6$
- Pre sadu 05 a 06 platí, že  $t = 1000$ ,  $a = 2$ ,  $b \leq 10^{18}$
- Pre zvyšné sady neplatia žiadne špeciálne podmienky.

<sup>3</sup>Možno si myslíte, že niektoré z týchto vecí som si vymyslel pre lepší epický tón rozprávky. Mýlite sa. Toto všetko sa naozaj nachádza v T2. A verte mi, je toho ešte oveľa viac.

<sup>4</sup>Z obavy vytvorenia čiernej diery sa neodvažujeme komprimovať predmety s objemom väčším ako  $10^{18}$ , pretože hmotnosť sa zachováva.

## Príklad

vstup

```
2
4 11 2
4 11 3
```

výstup

```
4
2
```

# Zadania kategórie T

Nájdete ich na našej stránke <http://www.ksp.sk/ulohy>. Nezapudnite sa na ne pozrieť, čakajú na vás ďalšie štyri zaujímavé úlohy rôznych obtiažností.