



## Úlohy 2. série letnej časti

Termín odoslania riešení tejto série je pondelok 30. mája 2016.

### 1. Zásuvky

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

V T2<sup>1</sup> sa počas dňa vyskytuje veľa programátorov a tí chcú, samozrejme, používať svoje notebooky. Notebooky musia byť zapojené do elektriny, keďže baterka sa im minula pri používaní počas prednášok. V T2 je našťastie obrovská predlžovačka s  $n$  zásuvkami.

Problém však nastal, keď si  $m$  KSP-ákov kúpilo nové Macbooky, ktoré majú najmodernejšie, revolučné a absolútne nepraktické zástrčky.<sup>2</sup> Sú o niečo širšie ako tie klasické a aj keď sa dajú strčiť do slovenskej zásuvky, na každej strane z nej trochu prečnievajú. Keď ich teda zastrčíte do predlžovačky, obe susedné zásuvky na predlžovačke sú blokované a nedá sa do nich vložiť žiadna iná zástrčka. To ale výrazne obmedzuje efektivitu spomínanej predlžovačky a KSP-ákov trápi, či si vôbec vedia všetci nabíjať svoje notebooky súčasne.

#### Úloha

Máme predlžovačku s  $n$  zásuvkami,  $m$  počítačov so širokými macovskými zástrčkami a  $k$  počítačov s normálnymi (úzkymi) zástrčkami. Zistíte, či sa všetkých  $m + k$  zástrčiek dá povkladať do predlžovačky. Normálna zástrčka zaberie práve jednu zásuvku a macovská zástrčka zaberie jednu zásuvku a zablokuje obe susedné zásuvky. Medzi dvoma macovskými zástrčkami však stačí mať jednu voľnú zásuvku.

#### Formát vstupu

Na prvom riadku dostanete  $n$  ( $1 \leq n \leq 10^9$ ) – počet zásuviek v predlžovačke. Na druhom riadku budú čísla  $m$  a  $k$  ( $0 \leq m, k \leq 10^9$ ) – počet macovských a počet normálnych zástrčiek.

#### Formát výstupu

Vypíšte **ano** ak sa notebooky zapojiť dajú a **nie**, ak to žiadnym spôsobom nejde.

#### Príklad

vstup	výstup
<input type="text" value="7"/> <input type="text" value="3 1"/>	<input type="text" value="ano"/>

*Rozložiť sa dajú napríklad takto: momonom (macovské – m, normálne – n, prázdna zásuvka – o). Všimnite si, že širokým koncovkám stačí medzi sebou iba jedna medzera.*

vstup	výstup
<input type="text" value="5"/> <input type="text" value="1 2"/>	<input type="text" value="ano"/>

*Napríklad rozloženie momon.*

vstup	výstup
<input type="text" value="2"/> <input type="text" value="1 2"/>	<input type="text" value="nie"/>

*Žiadnym spôsobom sa nepomestia, nemáme totiž ani dostatok zásuviek.*

vstup	výstup
<input type="text" value="4"/> <input type="text" value="2 1"/>	<input type="text" value="nie"/>

<sup>1</sup>Tajná KSP miestnosť na Matfyzě.

<sup>2</sup>Ak si ešte stále nie ste istí pri používaní slov zástrčka a zásuvka, [tento článok](#) vám určite pomôže.

## 2. Zapisovanie trpaslíkov

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

Všetci určite poznáte rozprávku o Snehulienke a siedmich trpaslíkoch. Snehulienka sa stratila v lese, našla domček v ktorom bývali trpaslíci, bol tam strašný neporiadok, a ona začala upratovať... Viete prečo nemali trpaslíci doma upratané? Nie, nebolo to preto, že by boli leniví... Trpaslíci toho majú len veľa na práci. Preto majú presne rozdelené, kto má čo spraviť, podľa poradia v ktorom sa v daný deň zobudili. Prvý chystá raňajky, druhý pripravuje stôl, tretí varí čaj, štvrtý upratuje... Aby sa v činnostiach vystriedali, a tiež, aby nemusel Spachtoš vždy vstávať ako prvý, každý deň vstávajú v inom poradí.

Aby trpaslíci vedeli zistiť, že sa už vystriedali všetky poradia, používajú nasledovný systém:

- Každý z trpaslíkov má svoju obľúbenú cifru od 0 do 9.
- Keď trpaslík ráno vstane, napíše svoju obľúbenú cifru na tabuľu, hneď za poslednú, ktorá tam bola. Takto vznikne každý deň na tabuli jedno číslo.
- Trpaslíci si počítajú súčet čísel, ktoré sa objavujú na tabuli – predtým, ako idú spať, pripočítajú číslo z tabule k súčtu.
- Keď je súčet dostatočne veľký, vystriedali sa všetky poradia a môžu sa začať striedať (a sčítavať) odznova.

Systém je dokonalý, no trpaslíci majú jeden problém. Nevedia, aký súčet má byť na tabuli po tom, čo sa vystriedali všetky poradia. Niekedy si tak nevšimnú, že sa už vystriedali a nasledujúce ráno nevstane nikto. Práve v takýto deň prišla Snehulienka a tá sa ako silná žena a nádejná programátorka rozhodla vyriešiť tento problém raz a navždy.

### Úloha

Pre jednu domácnosť trpaslíkov dostanete jedno číslo  $c_i$ , ktoré vzniklo na tabuli v prvý deň. Zistite, aký bude súčet čísel z tabule, keď sa vystriedajú všetky poradia trpaslíkov. Stručne povedané, zistite súčet všetkých permutácií cifier čísla  $c_i$ .

Úlohu vyriešte pre  $n$  rôznych domácností trpaslíkov.

### Formát vstupu

Na prvom riadku je číslo  $n$  ( $1 \leq n \leq 10^5$ ). Nasleduje  $n$  riadkov, v každom z nich je jedno číslo  $c_i$  ( $0 \leq c_i \leq 10^{12}$ ). Žiadne číslo iné ako 0 nezačína nulou.

### Formát výstupu

Vypíšte  $n$  riadkov, na  $i$ -tom riadku bude súčet permutácií cifier  $i$ -teho čísla zo vstupu. Každý súčet sa zmestí do 64 bitovej premennej (typu `Int64` v Pascale, `long long` v C++).

### Príklad

vstup	výstup
5	2
2	121
47	66
33	2664
750	113322
4247	

*Trpaslík s cifrou 2 býva sám, preto musí vždy vstať prvý. Pre trpaslíkov s ciframi 4 a 7 existujú dve poradia, ako môžu vstať (47 a 74). Trpaslíci v tretej domácnosti majú rovnaké obľúbené cifry, ale aj tak ich rozlišujeme a preto majú tiež dve možnosti.*

## 3. Záchrana princeznej

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

Jimi si nedávno kúpil nový herný počítač a jeho štúdium tým značne utrpelo. Je veľmi ťažké učiť sa, keď máte na výber z toľkých dobrých hier. Preto si vytvoril rozvrh, v ktorom má každú hodinu označenú ako pracovnú alebo hernú. Po čase hrania ale Jimiho prestalo baviť zabíjanie príšer a rozhodol sa splniť veľkú úlohu – zachráni princeznú ukrytú na hmlistom ostrove.

Hra však nepodporuje ukladanie počas plnenia misie a preto Jimi nemôže pracovať, kým nezachráni princeznú. Nechce si ale veľmi meniť rozvrh, preto len vyškrtne niektoré pracovné hodiny z rozvrhu tak, aby si vytvoril dostatočne dlhý súvislý úsek herného času na záchranu princeznej. Jeho štúdium je však už teraz riadne zanedbané, a preto chce obetovať čo najmenej zo svojho pracovného času.

## Úloha

Jimiho rozvrh vyzerá ako jedna dlhá postupnosť znakov P a H, ktoré označujú pracovné a herné hodiny.

Jimi si vypočítal, že na záchranu princeznej bude potrebovať  $c$  hodín hry.

Vašou úlohou je zistiť, koľko najmenej hodín práce vie vyškrtnúť z rozvrhu tak, aby mal **aspoň  $c$  hodín hry** v jednom kuse.

Môžete predpokladať, že herných hodín je v rozvrhu vždy aspoň  $c$ .

## Formát vstupu

V prvom riadku je číslo  $c$  – počet hodín potrebných na záchranu princeznej. V druhom riadku je rozvrh – reťazec znakov P a H dĺžky  $n$ .

Pre vstupy platí  $1 \leq c \leq n \leq 1\,000\,000$ .

## Formát výstupu

Vypíšte jedno číslo – najmenší počet hodín práce, ktoré treba vyškrtnúť z rozvrhu. Výstup ukončíte znakom nového riadku.

## Príklad

vstup	výstup
3 PPHHPPPHPHPHPP	2
<i>Upravený rozvrh vyzerá takto: PPHHPPPHHHPP.</i>	
vstup	výstup
4 HHPPPHPPHHHPH	1
vstup	výstup
2 HHPPHPP	0

## 4. Žiarivý zádrhel

kat. Z a O; 9 b za popis, 6 b za program

Kde bolo tam bolo, za bielym oblúkovým mostom a za dvoma veľkými semaforovými križovatkami, bola raz jedna škola. V tejto škole bola jedna trieda, a jej triedny učiteľ Ondrej si ju veľmi pochvaloval. Jej žiaci totiž chodili na informatické súťaže. Vyhrali na nich kopu trofejí, každú žiarivejšiu ako prechádzajúcu.

Jedného dňa si Ondrej povedal, že tak krásnych trofejí je v jeho kabinete škoda. Aby jeho žiaci mohli vidieť plody svojej práce, rozhodol sa, že ich vyloží na policičku v triede. Polička je však na všetky trofeje príúzká, Ondrej preto vystavil len tie najžiarivejšie.

Prišla jar, a spolu s ňou začalo do triedy prenikať čím ďalej tým viac slnečných lúčov. Polička s trofejami sa každé ráno trblietala krásnymi farbami. Ondrej si však po chvíli začal všimnúť aj jej temnú stránku. Trofeje žiarili tak výrazne, až to začalo všetkých v miestnosti oslepovať.

S tým treba niečo robiť! Ondrej teda vytiahol krabicu so zvyškom trofejí a začal dumať. Vie, ako najviac môže policička žiarieť aby žiakov ešte neoslepovala. Zároveň ju však nechce mať ani o kúsok matnejšiu, žiaci si predsa zaslúžia vidieť tie najžiarivejšie trofeje.

Ondrej má na výber  $n$  trofejí. Jeho žiaci vyhrávali čím ďalej tým žiarivejšie trofeje a tak  $i$ -ta trofej má žiarivosť  $i$ . Na policičku sa zmestí  $k$  trofejí, jej žiarivosť je súčtom žiarivostí jednotlivých trofejí. Najväčšia žiarivosť policičky, akú si môže Ondrej dovoliť, je  $s$ . Pomôžete mu vybrať tie správne trofeje?

## Úloha

Spomedzi čísel  $1, 2, \dots, n$  vyberte  $k$  takých, že ich súčet je presne  $s$ .

## Formát vstupu

Na vstupe dostanete tri celé čísla  $n, k, s$  oddelené medzerami.  $1 \leq n \leq 1\,000\,000$  – počet trofejí,  $1 \leq k \leq n$  – šírka policičky,  $1 \leq s \leq 2^{40}$  – ideálna žiarivosť policičky.

Na prácu s veľkými číslami<sup>3</sup> použite typ `long long` v C++ a `Int64` v Pascale.

<sup>3</sup>64-bitové čísla s rozsahom  $-2^{63}$  až  $2^{63} - 1$

## Formát výstupu

Na jediný riadok výstupu vypíšete  $k$  medzerou oddelených čísel – čísla (žiarivosti) trofejí, ktoré má Ondrej vyložiť na poličku. Na poradí vypísaných čísel nezáleží.

Ak dobrých výberov trofejí existuje viac, vypíšete ľubovoľný z nich. Ak žiadny dobrý výber trofejí neexistuje, vypíšete  $-1$ .

## Príklad

vstup	výstup
6 3 11	1 4 6
<i>Rovnako dobrý výber trofejí je napríklad aj 6, 2, 3.</i>	
vstup	výstup
5 2 10	-1

*Dve najžiarivejšie trofeje, ktoré môže Ondrej vystaviť sú 4 a 5, ani tie však nie sú dosť oslnivé.*

## 5. Otepľovanie v Absurdistane

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

Ako iste viete, v dôsledku globálneho otepľovania sa topia ľadovce a kvôli tomu stúpajú hladiny oceánov. V niektorých nízko položených krajinách, ako je napríklad Absurdistan, to potom spôsobuje problémy. So stúpajúcou hladinou mora totiž stúpa hladina podzemnej vody a – keďže Absurdistan má veľmi priepustné podložie – s ňou aj hladina nadzemnej vody. A tak postupne v rôznych kotlinách, údoliach a priehlbínach vznikajú nové jazerá, ktorým Absurdistanci vymýšľajú nové názvy.

### Úloha

Absurdistan má tvar obdĺžnika, ktorý si pre účely tejto úlohy rozdelíme na  $r \times s$  políčok. Každé políčko má svoju nadmorskú výšku<sup>4</sup> – celé číslo z rozsahu 0 až  $H$ . Hladina podzemnej vody je na začiatku 0. Každý rok hladina vody stúpne o 1 a všetky políčka, ktoré sa ocitli pod hladinou (ich nadmorská výška je ostro menšia ako výška hladiny), sa zatopia.

Každé zatopené políčko v Absurdistane je oficiálne súčasťou nejakého jazera. Pre jednoduchosť predpokladajme, že názvy Absurdistanských jazier sú nezáporné celé čísla. Každý rok sa v Absurdistane zide Kartografická Spoločnosť Patriotov a každému zatopenému políčku  $P$  určí, ktorého jazera je súčasťou, podľa nasledujúcich pravidiel:

- Každé políčko, ktoré bolo zatopené už aj rok predtým, ostáva súčasťou jazera, ku ktorému patrilo doteraz.
- Pre každé novozatopené políčko  $P$  skúsime nájsť políčko  $R$  také, že  $R$  bolo zatopené už aj pred rokom, z  $P$  sa dá dostať do  $R$  idúc iba po zatopených políčkach (pričom z jedného políčka vieme vždy ísť iba na políčka susediace stranou, ale nie na políčka susediace rohom) a cesta po vode z  $P$  do  $R$  je najkratšia možná. Políčko  $P$  sa stane súčasťou rovnakého jazera ako  $R$ . Ak je viacero možných políčok  $R$ , vyberieme také, ktorého jazero má najmenšie číslo.

Ak sa z  $P$  nedá po vode dostať na žiadne políčko, ktoré bolo zatopené už pred rokom, políčko  $P$  a všetky zatopené políčka, na ktoré sa z  $P$  dá dostať, budú prehlásené za novovzniknuté jazero.

- Novovzniknutým jazerám sa názvy priradia nasledovne: Prechádzame cez všetky políčka obdĺžnika, pričom postupujeme po riadkoch zhora nadol, a v rámci riadka zľava doprava. Vždy, keď nájdeme zatopené políčko patriace k nejakému novovzniknutému jazeru, ktoré ešte nemá názov, toto jazero nazveme najmenším nepoužitým nezáporným celým číslom.

Všimnite si, že keď sa v dôsledku stúpania hladiny dve jazerá zlejú do jednej súvislej vodnej plochy, oficiálne sú to stále dve rôzne jazerá s presne vymedzenými hranicami. Pre lepšie pochopenie týchto pravidiel si môžete pozrieť vzorový príklad s vysvetlením.

Vašou úlohou bude pre každé políčko zistiť číslo jazera, ku ktorému bude toto políčko patriť po  $H + 1$  rokoch, teda keď už bude celý Absurdistan zatopený.

<sup>4</sup>Pre jednoduchosť, za nadmorskú výšku budeme považovať nadmorskú výšku na začiatku zatápania, nie výšku nad aktuálnou hladinou mora. Nadmorská výška v našom chápaní sa teda počas zatápania meniť nebude.

## Formát vstupu

V prvom riadku vstupu sú dve kladné celé čísla  $r, s$  ( $r \cdot s \leq 250\,000$ ) – počet riadkov a počet stĺpcov obdĺžnika. V druhom riadku je jedno kladné celé číslo  $H$  ( $H \leq 250\,000$ ). Nasleduje  $r$  riadkov, každý z nich obsahuje  $s$  medzerami oddelených celých čísel z rozsahu 0 až  $H$  – nadmorské výšky jednotlivých políček.

## Formát výstupu

Vypíšte  $r$  riadkov a v každom z nich  $s$  medzerami oddelených čísel: čísla jazier, ku ktorým budú patriť jednotlivé políčka po úplnom zatopení Absurdistanu.

## Príklad

vstup	výstup
8 8	2 2 2 2 2 0 0 0
4	2 2 2 2 1 1 0 0
3 1 1 1 2 2 0 0	2 2 2 1 1 1 0 0
3 1 2 2 2 0 2 1	2 2 2 1 1 1 0 0
0 0 2 2 1 0 2 1	2 2 2 1 1 1 0 0
0 0 2 2 1 2 2 1	2 1 1 1 1 1 0 0
1 3 2 2 1 1 1 1	2 1 1 1 1 1 0 0
4 4 3 3 4 4 4 4	3 3 1 5 5 5 4 4
0 0 4 3 2 1 2 0	3 3 3 5 5 5 5 4
4 0 4 2 1 0 0 3	

Absurdistan bude po jednotlivých rokoch vyzerať nasledovne (modré oblasti sú zatopené, pričom svetlomodré sú novozatopené. Čísla v nezatopených políčkach udávajú nadmorskú výšku, čísla v zatopených udávajú číslo jazera):

Po 0 rokoch	Po 1 roku	Po 2 rokoch
3 1 1 1 2 2 0 0	3 1 1 1 2 2 0 0	3 2 2 2 2 2 0 0
3 1 2 2 2 0 2 1	3 1 2 2 2 1 2 1	3 2 2 2 2 1 2 0
0 0 2 2 1 0 2 1	2 2 2 2 1 1 2 1	2 2 2 2 1 1 2 0
0 0 2 2 1 2 2 1	2 2 2 2 1 2 2 1	2 2 2 2 1 2 2 0
1 3 2 2 1 1 1 1	1 3 2 2 1 1 1 1	2 3 2 2 1 1 0 0
4 4 3 3 4 4 4 4	4 4 3 3 4 4 4 4	4 4 3 3 4 4 4 4
0 0 4 3 2 1 2 0	3 3 4 3 2 1 2 4	3 3 4 3 2 5 2 4
4 0 4 2 1 0 0 3	4 3 4 2 1 5 5 3	4 3 4 2 5 5 5 3

  

Po 3 rokoch	Po 4 rokoch	Po 5 rokoch
3 2 2 2 2 0 0 0	2 2 2 2 2 0 0 0	2 2 2 2 2 0 0 0
3 2 2 2 1 1 0 0	2 2 2 2 1 1 0 0	2 2 2 2 1 1 0 0
2 2 2 1 1 1 0 0	2 2 2 1 1 1 0 0	2 2 2 1 1 1 0 0
2 2 2 1 1 1 0 0	2 2 2 1 1 1 0 0	2 2 2 1 1 1 0 0
2 3 1 1 1 1 0 0	2 1 1 1 1 1 0 0	2 1 1 1 1 1 0 0
4 4 3 3 4 4 4 4	4 4 1 1 4 4 4 4	2 1 1 1 1 1 0 0
3 3 4 3 5 5 4 4	3 3 4 5 5 5 4 4	3 3 1 5 5 5 4 4
4 3 4 5 5 5 5 3	4 3 4 5 5 5 5 4	3 3 3 5 5 5 5 4

## 6. Oddaľovanie roboty

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

Bob by mal pracovať na dizertačnej práci, ale veľmi sa mu nechce. Keďže do ukončenia PhD *zostáva ešte čas*, Bob preferuje iné činnosti ako napríklad hranie frisbee.

Každý Bobov rok vyzerať tak, že mu školiteľ dá  $n$  rôznych problémov, ktoré by mal vyriešiť v rámci dizertačnej práce. Aby školiteľ Boba motivoval,<sup>5</sup> každá úloha má svoj termín, dokedy ju má urobiť. Bob sa ale statočne bráni a má celkom jednoduchú stratégiu. Najprv sa chce čo najdlhšie flákať (venovať sa iným činnostiam ako dizertačke), a až potom začne robiť na úlohách.

Navyše sa mu podarilo zistiť, že jeho školiteľ je príliš mäkký. Bob si tak môže beztrestne dovoliť nespraviť najviac  $k$  úloh. Ako najdlhšie sa môže Bob zo začiatku flákať pri takejto stratégii?

<sup>5</sup>inak povedané, aby zabránil jeho úplnému flákaniu sa

## Úloha

Máme zadaných  $n$  úloh. Pre každú úlohu vieme, koľko času Bob potrebuje na to, aby ju splnil –  $t_i$  a dokedy ju má splniť – deadline  $d_i$ . Vašou úlohou je zistiť, ako najdlhšie sa môže na začiatku flákať, pričom si môže dovoliť nespraviť najviac  $k$  úloh. Bob vie pracovať v jednom čase najviac na jednej úlohe. Keď Bob začne pracovať na úlohe, celú ju spraví naraz.

### Formát vstupu

V prvom riadku sú dve celé čísla oddelené medzerou  $1 \leq n \leq 3000$  a  $0 \leq k < n$ . V ďalších  $n$  riadkoch sa nachádzajú dvojice celých čísel oddelených medzerou  $1 \leq d_i \leq 10^9$  a  $1 \leq t_i \leq \min(d_i, 10^6)$ , pričom  $d_i$  je deadline do ktorého treba splniť  $i$ -tu úlohu a na jej dokončenie treba  $t_i$  času.

V jednej sade vstupov zo štyroch bude platiť  $k = 0$ .

### Formát výstupu

Vypíšte jedno nezáporné celé číslo – ako dlho sa môže Bob na začiatku flákať, aby nestihol najviac  $k$  termínov. Bob sa chce najskôr čo najdlhšie flákať, a potom, až keď je to nevyhnutné, začne robiť. Ak sa úloha nedá splniť, vypíšte **Neda sa to stihnúť**.

### Príklad

vstup

```
2 0
10 5
7 1
```

výstup

```
4
```

Bob začne s druhým termínom v čase 4. Dokončí ho v čase 5, a potom pracuje na prvom, ktorý dokončí v čase 10.

vstup

```
3 1
10 3
8 2
6 2
```

výstup

```
5
```

Bob si povie, že zmešká tretí termín a začne druhým v čase 5 a dokončí ho v čase 7. Potom začne robiť na prvej úlohe. Môžete si všimnúť, že ak by Bob nespravil prvú úlohu, mal by celkovo viac voľného času, lenže jemu ide len o to, aby prácu čo najviac oddialil.

vstup

```
2 0
10 10
2 2
```

výstup

```
Neda sa to stihnúť.
```

## 7. Obézni brankári

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

„Čížiček, čížiček, vtáčik maličký, povedz nám čížiček, ako brániť gól...”

Nuž, Joža táto filozofická otázka zaujala tiež. Až natoľko, že v jednom interview pre prestížny časopis DRB++, ktorého novinári pravidelne vyhadávajú tie najvplyvnejšie osobnosti, povedal Jožo tento známy citát: „Prečo monštruózne obézni ľudia nie sú hokejoví brankári?”. Táto otázka už bola citovaná množstvom filozofov. Aj vďaka tomu je Jožo považovaný za jedného z najvplyvnejších mysliteľov modernej filozofie.

Dnes Joža trápí podobná, avšak trochu iná otázka. Na Wikipédii sa dočítal, že existujú trpaslíci aj obri. A aj trpaslíci, aj obri, môžu byť monštruózne obézni. Je však jasné, že monštruózne obézny trpaslík je oveľa menší, než monštruózne obézny obor. Jedna vec ich však všetkých spája a síce to, že sú rovnako širokí, ako vysokí.

A keďže Jožo sa vo svojom myslení nikdy nezastaví a vždy ide hlbšie a hlbšie, teraz ho zaujíma takáto otázka: koľkými spôsobmi je možné vyplniť veľkú hokejovú bránku monštruózne obéznymi ľuďmi tak, aby sa do nej nedal streliť gól?

## Úloha

Monštruózne obézni ľudia sú štvorcoví a majú rozmery  $1 \times 1$ ,  $2 \times 2$ ,  $3 \times 3$ , alebo  $4 \times 4$  metre.

Máme veľkú hokejovú bránu s rozmermi  $4 \times n$  metrov. Zistite, koľkými spôsobmi je možné ju úplne vyplniť monštruózne obéznyimi ľuďmi (môžu stáť na sebe) tak, aby sa žiadni neprekrývali a aby nebolo ani trochu voľného miesta pre puk.

Keďže výsledok môže byť pomerne veľký, vypíšte jeho zvyšok po delení  $m$ .

### Formát vstupu

Na vstupe sú na jednom riadku čísla  $n$  a  $m$  oddelené medzerou. Platí  $1 \leq n \leq 10^{16}$  a  $1 \leq m \leq 10^7$ .

### Formát výstupu

Vypíšte jeden riadok a na ňom jedno číslo: počet možností, ako zaplniť bránu  $4 \times n$  monštruózne obéznyimi ľuďmi modulo  $m$ .

### Príklady

vstup	výstup
3 47	13
vstup	výstup
7 1000	29

## 8. Opitý za volantom

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

V Krajine samých pijanov sú čoraz častejšie pristihnutí vodiči za volantom pod vplyvom alkoholu. To, či je to spôsobené lepšou kvalitou policajnej hliadky, alebo či to súvisí s parádnym hitom vinárskeho trhu s názvom „pangalaktický dunihlav“, je pre všetkých záhadou.

Situácia v krajine ale začína byť vážna – ľudia sa štítia pitia alkoholu, pretože sa boja, že by mohli skončiť za volantom následkom čoho začína celá ekonomika upadať.

Časom si policajné hliadky všimli, že sa vodiči často sťažujú na dopravné značky – buď ich nie je dobre vidno, alebo sú im otočené chrbtom, alebo ich v spoločensky unavenom stave nestihnú zaregistrovať<sup>6</sup>.

Prišli teda s nasledujúcim famóznym nápadom: každý vodič bude mať úžasné zariadenie. Vodič sa môže v prípade potreby zariadenia spýtať na všetky dopravné obmedzenia, ktoré sa ho momentálne týkajú. Stačí stlačiť veľké priateľské tlačidlo *PODLIEHAM PANIKE* a zariadenie vypíše zoznam všetkých predpisov.

### Úloha

Na jednej dlhej ceste platia na rôznych úsekoch rôzne dopravné obmedzenia. Každé dopravné obmedzenie je určené jeho začiatkom, koncom a poradovým číslom. Pozície na ceste sú reprezentované celými číslami.

Ak sa vodič nachádza na pozícii  $p$ , dopravné obmedzenie so začiatkom  $l$  a koncom  $r$  sa ho týka práve vtedy, keď  $l \leq p < r$ .

Dostanete zoznam začiatkov a koncov  $n$  obmedzení a následne  $q$  otázok. Otázka je jedno číslo – pozícia vodiča na ceste. Pre každú otázku vypíšte zoznam všetkých obmedzení, ktoré sa vodiča týkajú.

Otázky je nutné spracovať online – nemôžete spracovať ďalšiu otázku, kým nezodpoviete tú aktuálnu.

### Formát vstupu

Na prvom riadku vstupu je počet dopravných obmedzení –  $n \leq 5 \cdot 10^5$ . Nasleduje  $n$  riadkov, na  $i$ -tom z nich je popis dopravného obmedzenia s poradovým číslom  $i$ .

Popis dopravného obmedzenia pozostáva z dvoch celých čísel  $l, r$  – jeho začiatok a koniec. Platí  $0 \leq l, r \leq 2n$ .

Nasleduje číslo  $q \leq 2n$  – počet otázok. Na každom z ďalších  $q$  riadkov je jedno celé číslo  $p$ . Nech  $m$  je aktuálna maska (ktorej výpočet je popísaný v nasledujúcom odseku). Potom pozícia vodiča je  $p \oplus m$  (bitový xor  $p$  a  $m$ , v C++  $p \wedge m$ ).

### Formát výstupu

Pre každú otázku  $p$  vypíšte najprv počet dopravných obmedzení  $d$ , ktoré sa vodiča týkajú. Následne vypíšte do toho istého riadku čísla  $a_1, a_2, \dots, a_d$  – poradové čísla týchto dopravných obmedzení v ľubovoľnom poradí. Čísla oddeľte jednou medzerou, a za posledným číslom ukončíte riadok.

Pre prvú otázku je hodnota masky  $m = 0$ .

<sup>6</sup>čo je samozrejme chyba dopravnej značky

Pre ostatné otázky vypočítame hodnotu masky nasledovne: nech  $m'$  je hodnota masky na predchádzajúcej otázke, a nech odpoveď na ňu bola  $d a_1 a_2 \dots a_d$ . Potom aktuálnu hodnotu masky vypočítame ako  $m = m' \oplus a_1 \oplus a_2 \oplus \dots \oplus a_d$ .

**Je zaručené, že počet čísel na správnom výstupe nebude presahovať  $5 \cdot 10^5 + q$ .**

### Príklady

vstup

```
4
5 7
3 4
1 2
0 6
8
5
7
6
7
3
3
3
2
```

výstup

```
2 0 3
1 3
1 0
0
2 1 3
2 2 3
1 3
1 3
```

## Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trochu ťažšia ako kategória O, ale mnohí z vás ju určite zvládnu).

Body z tejto kategórie sa mierne zohľadňujú aj pri výbere tímu, ktorý pôjde súťažiť na Medzinárodnú olympiádu v informatike, tento rok v Rusku.