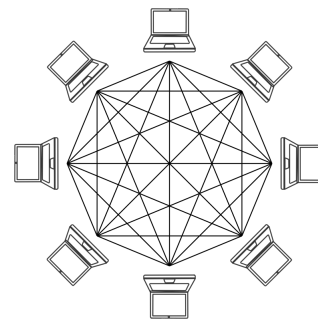


## Úloha č. 2

### Routování



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“.

„Tohle je divné,“ řekl Gandalf nad důkladně rozebranou obludou, „vždyť má záporné TTL (Time To Live).“ „Co je to Gandalfe?“ zeptal jsem se.

„Ne každá síť je správně nakonfigurovaná. Některé obsahují routovací smyčky, ve kterých by pakety mohly bloudit donekonečna. Abychom tomu předešli, každý packet má tzv. TTL. Je to hodnota, která se na každém uzlu zmenší o jedna. Když dosáhne nuly, je packet zahozen. Pokud nastavíme hodnotu rozumně, při běžném routování se TTL nikdy nevyčerpá. Pokud ale packet spadne do smyčky, velmi rychle mu TTL vyprší a je zahozen.“

„Takže pokud měla ta obluda zápornou hodnotu, při každém přesunu mezi uzly se stávala akorát zapornější a zapornější,“ přemýšlel nahlas Aragorn. „To je ale záporák,“ řekl Gimli, „huehehehe.“

„To je možné. Každopádně, než přijdeme na to, odkud se záporáci berou, měli bychom alespoň upravit naši část Systému, aby nebyli záporáci tak záporní, jak jsou. Pro začátek bych alespoň rozdělil síť do podsítí.“

„Pro takové rozdělení Systému existují prastará pravidla,“ pokračoval Gandalf. „Všechny podsítě budou mít stejnou masku, která bude mít co **největší hodnotu**. Maska je libovolné 32bitové číslo, v němž jedničky značí, na jakých bitech se musí shodovat adresy všech uzlů patřících do jedné podsítě (každý uzel má svou IP adresu a zařadíme jej do právě jedné podsítě).“<sup>1</sup>

„Co to ale vlastně udělá?“ zeptal jsem se.

„Když pakety cestují napříč podsítěmi, tak jim to trvá,“ odpověděl Aragorn, „čím jsou dvě podsítě dál od sebe (čím jsou bity, které je charakterizují, rozdílnější), tím déle přecházejí pakety jejich hranice.“

„Kéž by to bylo tak jednoduché!“ odvětil Legolas. „Ale stejně jako dává mému ostřížímu zraku Systém limit, existuje nejmenší změřitelná časová jednotka pro přechod přes hranice.“

„To je pravda,“ přitakal Aragorn, „rozdíl mezi časy na překročení hranic podsítí, které se liší o 1, 2, resp. 3 bity, je moc malý, Systém proto tyto tři časy považuje za ekvivalentní; to je základní jednotka času, tedy 1. Další časy jsou rovny polovině počtu rozdílných bitů, zaokrouhлено dolů.“

„Znamená to, že někdy je nejrychlejší cesta oklikou přes třetí podsítí?“ zeptal jsem se po chvíli.

## Vstup

Na prvním řádku je počet zadání  $t$  ( $1 \leq t \leq 30$ ). Každé zadání začíná dvojicí čísel  $n, k$  ( $1 \leq n \leq 200000$ ,  $1 \leq k \leq 500$ ), kde  $n$  je počet uzlů a  $k$  je maximální počet podsítí, do kterých je máme rozdělit. Následuje  $n$  řádků, na každém je IP adresa příslušného uzlu, tedy 4 čísla v rozmezí 0 až 255 oddělená tečkou. Každé z těchto 4 čísel určuje 8 bitů adresy.

<sup>1</sup>Pozor, běžně (ve skutečném světě) tvoří masku sled jedniček následován sledem nul. Systém však povoluje i masky, kde se bity 1 a 0 střídají. Tedy 1010 je validní začátek masky.

## Výstup

Pro každé zadání vypiš dva řádky. Na jeden řádek vypiš masku, dle které jsi vytvořil podsítě, ve stejném formátu jako IP adresy na vstupu. Na druhý vypiš součet časů, které je při cestování mezi libovolnými dvěma podsítěmi možné ušetřit cestou přes další podsítě (pro každou dvojici přičti ušetřený čas právě jednou, zpáteční cestu nepočítej).

Nižší počet bodů Ti bude udělen, vyřešíš-li pouze první část úlohy, do výstupu dej v tom případě pouze masky, jednu na řádek.

### Vstup

```
3
4 2
192.168.1.0
192.196.196.0
128.1.1.0
42.168.0.42
2 1
192.168.1.1
64.168.1.1
6 3
3.248.15.8
0.32.96.103
3.222.215.208
0.0.54.16
0.93.84.80
0.88.99.215
```

### Výstup

```
191.18.58.255
0
127.255.255.255
0
255.216.0.0
1
```