

## 2a. Start vesmírné lodi

**2 body**

**Formulace problému:** Patříte mezi první kolonizátory Marzu. Došlo ke kritickému selhání systému a vy musíte co *nejrychleji* odletět pryč. Při startu meziplanetární lodi musíte zapojovat jednotlivá spojení mezi řídicími uzly, čímž dochází k jejich aktivaci. Postupně musíte aktivovat všechny řídicí uzly. Nicméně různé spoje se aktivují *různě dlouho* a pro aktivaci všech řídicích uzlů nemusíte zapojit všechny spoje! Musíte se rozhodnout, které spoje aktivujete a které ne. Ale pozor! Pokud by jste novým spojem propojili již dříve aktivované uzly, tak je přetížíte a dojde ke zkratu a celá loď exploduje. A to opravdu nechcete.

**Název spustitelného souboru:** launch

**Vstup:** Nejprve dostanete seznam řídicích uzlů

```
CPU: motory, navigace, komunikace, mustek, kyslik
motory - kyslik: 15s
navigace - komunikace: 90s
mustek - kyslik: 20s
motory - navigace: 50s
komunikace - mustek: 35s
kyslik - komunikace: 15s
```

**Výstup:** Výstupem seznam aktivovaných linek.

```
motory - kyslik
navigace - komunikace
mustek- kyslik
motory - navigace
```

## 2b. Důležitá města a silnice

**2 body**

**Formulace problému:** Jako evropský autobusový dopravce jste přebíral konkurenční společnosti a tak máte spoustu zdvojených spojů. Zadal jste svým manažerům úkol, aby navrhli optimalizaci celé sítě linek a některé nepotřebné spoje vyřadili. Dostal jste od nich níže uvedený návrh. Z vlastní zkušenosti víte, jak je těžké získat smlouvy s městy a povolení jezdit po určitých dálnicích. A tak potřebujete v návrhu najít město (dálnici), jehož výpadek ze sítě (kvůli neuzavření smlouvy) by způsobil, že vaše společnost není schopna převést zákazníka z libovolného vámi obhospodařovaného města do jiného vámi obhospodařovaného města, protože nemůžete daným městem (resp. po dané dálnici) projet. Možná tokové město neexistuje, možná je jen jedno, možná jich je více. Obdobně platí pro dálnice.

**Název spustitelného souboru:** weakness

**Vstup:** Textový soubor obsahuje seznam měst a existující linky

```
City: Praha, Berlín, Paříž, Vídeň, Londýn
LS2021: Praha -> Berlín -> Paříž -> Praha
ZS2020: Praha -> Londýn -> Paříž -> Vídeň
AB111: Vídeň -> Londýn -> Paříž -> Berlín
XYZ007: Praha -> Paříž -> Vídeň
```

**Výstup:** Výstupem algoritmu bude nejprve seznam linek, jejichž výpadek by vedl k rozpadu sítě a následně i seznam kritických transformátorů.

```
Praha
ZS2020: Praha -> Londýn
XYZ007: Paříž -> Vídeň
```

## 2c. Únik z vesmírné lodi

**2 body**

**Formulace problému:** V předchozím problému, kdy jste startovali vesmírnou loď, jste udělali chybu a došlo ke zkratu a musíte loď opustit. Nicméně, neznáte cestu ven a musíte ji najít. Loď je obrovská a obsahuje spoustu sekcí, které jsou rozdělené tlakovými dveřmi. Bohužel v tak složitém prostoru nemůžete uplatnit nějaký jednoduchý systém procházení, jako je pravidlo levé či pravé ruky.

**Název spustitelného souboru:** escape

**Vstup:** Začínáte v sekci **mustek**, a snažíme se dostat do sekce “unikový\_modul” Seznam sekcí a průchodu mezi nimi.

```
Sections: mustek, motory, navigace, komunikace, kyslik,  
unikovy_modul  
mustek - motory  
mustek - kyslik  
motory - kyslik  
motory - navigace  
kyslik - komunikace  
komunikace - navigace  
navigace - unikovy_modul
```

**Výstup:** Seznam průchodů, kterými jste při hledání cesty prošli.

```
mustek -> motor  
motor -> kyslik  
kyslik -> mustek  
mustek -> kyslik  
kyslik -> komunikace  
komunikace -> navigace  
navigace -> motor  
motor -> navigace  
navigace -> unikovy_modul
```

## 2d. AVL Strom

**3 body**

**Formulace problému:** Na vstupu programu máte textový soubor s řadou číslic. Vaším úkolem je tento soubor postupně načítat a ukládat je do AVL stromu. V každém kroku vypíšte AVL strom.

**Název spustitelného souboru:** avltree

**Vstup:** Textový soubor, ve kterém je na každém řádku uvedeno jedno přirozené číslo. Může vypadat například takto:

```
5
15
9
5
155
```

**Výstup:** Výstupem bude seznam čísel reprezentující strukturu AVL stromu přepsanou pomocí metody procházení do šířky, kdy úrovně budu odděleny znakem | a prázdné místo (chybějící potomek) bude reprezentován znakem \_. Výpis bude končit poslední úrovní. Pro vstupní soubor z příkladu by pak výsledek programu mohl vypadat následovně:

```
5
5 | _ 15
9 | 5 15
9 | 5 15 | _ 5 _ _
9 | 5 15 | _ 5 _ 155
```