Lehký příklad: Anagramy (6 bodů)

Napište program, který v zadaném souboru hledá dvojice anagramů. Anagram (přesmyčka) je textový řetězec, který z původního řetězce vznikne tak, že se použijí všechna původní písmena, ale změní se jejich pořadí. Například: 'debit card' → 'bad credit'.

Vstup jméno souboru zadáno jako první argument příkazové řádky. Na každém řádku tohoto souboru je jeden textový řetězec (string), který může obsahovat malá/velká písmena a mezery.

Výstup Seznam dvojic celých čísel (vždy jedna dvojice na řádku), nebo slovo 'None' vytištěné na standardní výstup.

- Dvojice celých čísel označuje indexy řádků ve vstupním souboru, které jsou anagramy. Řádky jsou číslovány od nuly.
- Dvojice jsou seřazeny vzestupně dle prvního indexu, v případě shodnosti prvních indexů jsou seřazeny vzestupně dle druhého indexu (např. '10 11', '10 12', '10 13', '11 12', '11 13' ...) (viz příklad IV).
- První číslo ve dvojici je vždy menší než druhé.
- Je nutné vypsat všechny dvojice splňující výše uvedené podmínky.
- V případě, že vstupní soubor neobsahuje ani jednu dvojici anagramů, vypište 'None'.

Odevzdání Program nahrajte do Bruta v souboru **anagram.py** — úloha **ZK1-easy**.

- Je zaručeno, že vstupní soubor existuje a obsahuje alespoň jednu řádku.
- Je zaručeno, že každá řádka obsahuje pouze kombinaci malých/velkých písmen a mezer.
- Mezery se při určování anagramů neuvažují.
- Při určování anagramů je třeba rozlišovat velikost písmen (anagramy jsou case-sensitive).
- Při řešení můžete použít libovolné funkce jazyka Python, včetně standardních knihoven dostupných v systému Brute.
- Nápověda: složitost řešení je vzhledem k počtu řetězců (a jejich délce) kvadratická nebo lepší.

Příklady anagramů:

- 'abc', anagramem je např. 'bca', 'cab', 'c a b', ' a c b' (mezery se neuvažují)
- 'abc' a ' a a b c ' nejsou anagramy (nesouhlasí počet písmen)
- 'Ac' a 'CA' nejsou anagramy (case-sensitive)
- 'forty five', příklad anagramu: 'over fifty' nebo 'overfifty' nebo 'fiftyover'

Bodování:

Popis části	Počet testů	Timeout	Max. bodů	Hodnocení
Krátké I (do 4 řádků, bez mezer)	50	celkem 2 s	**1**	0.02b/test
Krátké II (do 4 řádků, včetně mezer)	50	celkem 2 s	**1**	0.02b/test
Slovník (do 10 řádků, anglická slova)	50	celkem 2 s	**1**	0.02b/test
Náhodné (do 20 řádků, náhodné řetězce do 100 znaků)	100	celkem 2 s	**2**	0.02b/test
None (do 50 řádků, řešením je 'None')	10	celkem 2 s	**1**	0.1b/test

Jednotlivé části testů budou spuštěny až po dosažení plného počtu bodů v předchozí části, např. testování příkladů typu 'None' bude provedeno až poté, co všechny předchozí testy dopadnou správně.

Příklad I Volání `python3 anagram.py abc.txt`
Obsah souboru `abc.txt`

```
abc
Abc
a c b
   cA b
```

Výstup

```
0 2
1 3
```

Příklad II Volání `python3 anagram.py simple.txt`
Obsah souboru `simple.txt`

```
tacts
lemur
bonny
argue
asttc
ruelm
onbyn
gaeru
```

Výstup

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

Příklad III Volání `python3 anagram.py long.txt`
Obsah souboru `long.txt`

```
Cockpit
Destroy
Firedog
Concern
Fiction
Postbox
Peanuts
Percent
Network
```

Výstup

```
None
```

Příklad IV Volání `python3 anagram.py long.txt`
Obsah souboru `long.txt`

```
Seaward
Pension
Thirsty
Unlevel
Helpful
Dynamic
Abalone
Kennedy
Equinox
Comfort
AbalonePensionDynamicHelpfulComfortSeawardKennedyUnlevelThirsty Equinox
KennedySeawardHelpfulDynamic EquinoxComfortUnlevelPension  ThirstyAbalone
ThirstyAbaloneUnlevelKennedySeaward  DynamicHelpful EquinoxComfortPension
AbaloneDynamicSeawardPensionEquinoxThirstyComfortUnlevel HelpfulKennedy
KennedyAbalone ThirstyComfortPension DynamicEquinoxSeaward UnlevelHelpful
AbaloneDynamicThirstyKennedyUnlevelEquinoxHelpful Seaward Pension Comfort
ThirstyAbalonePension KennedyDynamicEquinox  HelpfulUnlevelSeawardComfort
KennedyUnlevelEquinoxComfortSeawardAbalonePension HelpfulDynamic Thirsty
```

Výstup

```
10 11
10 12
10 13
10 14
10 15
10 16
10 17
11 12
11 13
11 14
11 15
11 16
11 17
12 13
12 14
12 15
12 16
12 17
13 14
13 15
13 16
13 17
14 15
14 16
14 17
15 16
15 17
16 17
```

Komentář Jeden řetězec může samozřejmě být anagramem několika jiných řetězců. V tomto případě řádek 10 (číslujeme od nuly) je anagramem všech dalších řádků (11,12,...,17) a stejně tak řádek číslo 11 je anagramem těch následujících. Jelikož vypisujeme tak, že první číslo je menší než druhé, vypíšeme pouze kombinaci '10 11', zatímco '11 10' ne.

Těžký příklad: Erdősovo číslo (14 bodů)

Slavný mad'arský matematik Paul Erdős (1913–1996) napsal mnoho vědeckých publikací, často ve spolupráci s jinými matematiky. Na jeho počest bylo zavedeno tzv. Erdősovo číslo (označme ho $E(x)$ pro osobu x), které vyjadřuje, jak jsou matematici "autorsky" vzdáleni od P. Erdőse. Platí, že P. Erdős má E(Erdős) = 0. Přímí spoluautoři jeho článků mají číslo 1, přímí spoluautoři přímých spoluautorů mají číslo 2 atd. Pokud má osoba x publikaci s autory y a z, její Erdősovo číslo se odvozuje od toho menšího, tedy: $E(x) = \min(E(y), E(z)) + 1$. Cílem této úlohy je načíst soubor s údaji o spoluautorství a určit $E(x)$ pro zadané x.

Vstup jméno souboru zádáno jako první argument příkazové řádky; jméno hledané osoby jako druhý argument příkazové řádky.

Každý řádek vstupního souboru obsahuje vztah mezi autory ve formátu:

```
jmeno1 jmeno2
```

(tento vztah znamená, že `'jmeno1'` a `'jmeno2'` mají společnou publikaci)

- Jedna řádka vždy obsahuje jméno 'erdos'.
- Jména jsou tvořena pouze malými písmeny a jsou oddělena mezerou.
- Jméno zadané osoby (2. arg. příkazové řádky) je taktéž tvořeno malými písmeny.

Výstup celé číslo na standardní výstup, nebo `'None'`.

- Pro zadanou osobu x určete její $E(x)$ a vypište jako celé číslo.
- V případě, že pro tuto osobu nejde $E(x)$ určit, vypište `'None'`.

Odevzdání Program nahrajte do Bruta v souboru **erdos.py** — úloha **ZK1-hard**.

- Je zaručeno, že vstupní soubor existuje a obsahuje alespoň jednu řádku.
- Není zaručeno, že hledaná osoba x se vyskytuje ve vstupním souboru.
- Nápověda: Erdősovo číslo je vzdálenost uzlů v grafu.

Bodování

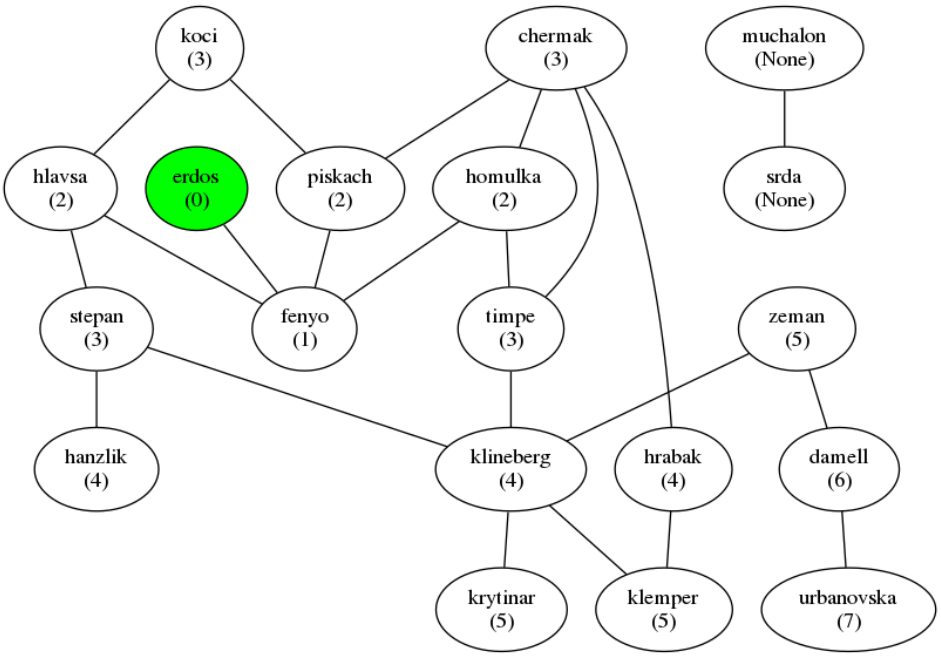
Popis části	Počet testů	Timeout	Max. bodů	Hodnocení
Soubory s max. 10 autory, řešení existuje	20	celkem 2 s	**2**	0.01b/test
Soubory s max. 20 autory, řešení existuje	20	celkem 2 s	**2**	0.01b/test
Soubory s max. 50 autory, řešení existuje	20	celkem 2 s	**4**	0.02b/test
Soubory s max. 100 autory, řešení existuje	20	celkem 2 s	**4**	0.02b/test
Soubory s max. 100 autory, řešení None	20	celkem 2 s	**2**	0.01b/test

Jednotlivé části testů budou spuštěny až po dosažení plného počtu bodů v předchozí části, např. testování příkladů typu 'None' bude provedeno až poté, co všechny předchozí testy dopadnou správně.

Soubor `autori.txt`

```
erdos fenyo
zeman klineberg
zeman damell
koci hlavsa
koci piskach
muchalon srda
chermak homulka
chermak timpe
chermak hrabak
chermak piskach
homulka timpe
homulka fenyo
timpe klineberg
hlavsa stepan
hlavsa fenyo
hrabak klemper
stepan klineberg
stepan hanzlik
klineberg krytinar
klineberg klemper
piskach fenyo
damell urbanovska
```

Zobrazení autorské spolupráce ze souboru `autori.txt`. Čísla pod jménem označují Erdősovo číslo, slovo 'None' vyjadřuje neexistenci tohoto čísla v případech, kdy osoba není autorsky spojena s panem Erdősem.



Příklad I Volání `python3 erdos.py autori.txt koci`
Výstup

```
3
```

Komentář: protože autor `koci` má společnou publikaci s `piskach`, `piskach` má publikaci s `fenyo` a ten přímo s `erdos`. Autor `fenyo` je tedy přímým spoluautorem `erdose` a proto má E(fenyo)=1, E(piskach)=2 a konečně E(koci)=3.

Příklad II Volání `python3 erdos.py autori.txt muchalon`
Výstup

```
None
```

Komentář: autor `muchalon` má společný článek pouze s `srda`, ale ani jeden z nich není autorsky spojen se jménem `erdos`.