

Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

39. ročník

KSP-Z

Říjen 2026

Právě se díváte na leták druhý série 39. ročníku **KSP-Z**, neboli **Korespondenčního Semináře z Programování, Začátečnické** kategorie. Zapojit se může **každý středoškolák i základoškolák**.

V průběhu roku vydáme několik dalších sérií úloh podobných této. Pokud budete mít jakoukoliv otázku, neváhejte se zeptat. Kontaktní adresy najdete v patičce na konci letáku. Přejeme hodně štěstí!

Termín série: **neděle 29. listopadu ve 32:00** (tedy další ráno v 8:00), praktické úlohy za třetinu bodů až do 6. prosince

Obsah série: **3 praktické úlohy** (značené ) – K těmto úlohám je nutné napsat program (v libovolném vhodném jazyce), stáhnout si z našeho webu vstupní data a odevzdat odpovídající výstup.
1 teoretická úloha (značená ) – U této úlohy nás zajímá hlavně slovní popis řešení, ve kterém byste měli zdůvodnit jeho funkčnost a ideálně nás i přesvědčit o jeho efektivitě.

Odevzdávání: Přes web na adrese <https://ksp.mff.cuni.cz/z/odevzdavatko/>



Zadání druhé série začátečnické kategorie 39. ročníku KSP

Po horkých dnech v Itálii se Kevin a Sára vydali do Norska. Problémy je dobehly již na prvním ubytování v Oslu. Pomozte jim několik z nich vyřešit, aby se mohli konečně věnovat turistice a poznávání norské kultury.

39-Z2-1 Vikingové šifrují 8 bodů

Sára s Kevinem se konečně dostali ke svému ubytování v Oslu, kam vyrazili na Erasmus+ pobyt. Po té předlouhé cestě vlakem už se jim chtělo opravdu spát, a tak Sára z ospalosti uchopila kliku, vzala za ni... a nic. Co budou jen dělat? No přece seženou klíč, to dá rozum.

Vtom si oba všimli velkého papíru na zdi, který byl celý popsaný nějakým textem. Kevinovi se na papíru něco nezдалo, a tak vytasil metr a začal papír měřit. „Hm, ten papír je větší než A0,“ poznamenal. Poté, co se oba dodivili této nestandardní skutečnosti, si všimli ještě dvou věcí: papíru, tentokrát už běžného formátu B4, a jakési krabičky s kódovým zámekem a piktogramem klíče. Místo obvyklých tří nebo čtyř číslic měl zámek na krabičce číslic mnoho – stovky, možná tisíce. Na papíru (tom formátu B4) se dozvěděli, že potřebují vyřešit Vikingskou šifru a výsledkem bude několik čísel, která otevrou zámek na boxu a umožní jim dostat se ke klíči uvnitř, s jehož pomocí už se dostanou do bytu.

Jak vlastně Vikingská šifra funguje? Dostanete zadané jedno krátké slovo S a jeden dlouhý text T . Vaším úkolem je v textu T najít a zakroužkovat písmena tak, aby když je přečtete za sebou zleva doprava, poskládala přesně slovo S . Výsledný kód k zámku pak vytvoříte tak, že si jednoduše zapíšete pozice (pořadová čísla) všech zakroužkovaných písmen v textu T . Pozice počítáme od nuly, takže první znak textu má číslo 0, druhý číslo 1 a tak dále.

Toto je praktická open-data úloha. V odevzdávacím systému si necháte vygenerovat vstupy a odevzdáte příslušné výstupy. Záleží jen na vás, jak výstupy vyrobíte.

Formát vstupu: Na prvním řádku dostanete čísla N a M – počet znaků slova S a počet znaků textu T . Na druhém řádku dostanete slovo S a na třetím řádku text T .

Slibujeme, že S i T se skládají pouze z písmen anglické abecedy.

Formát výstupu: Pro každý znak S uveďte pozici stejného

znaku z T tak, aby pozice znaků byly ve stejném pořadí jako v S . Všechny pozice uveďte na jeden řádek a oddělte mezerami. Znaků číslujeme od nuly. O nalezených pozicích nepožadujeme nic dalšího; nemusí se např. jednat o první možné výskyty.

Ukázkový vstup:

3 11

abc

bacbobrocac

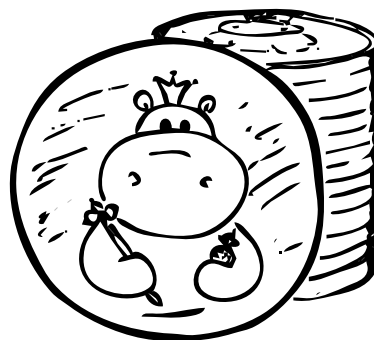
Ukázkový výstup:

1 3 8

39-Z2-2 Magické mince 10 bodů

 Sára se v Oslu vydala na procházku do Vigelandsparku. U vchodu potkala norské studenty Norberta a Olafa, kteří právě hráli zvláštní hru s mincemi. Přidala se k nim, ovšem brzy zjistila, že Norbert hraje až podezřele dobře. Pomozte Sáře, aby ho porazila.

Toto je praktická open-data úloha. V odevzdávacím systému si necháte vygenerovat vstupy a odevzdáte příslušné výstupy. Záleží jen na vás, jak výstupy vyrobíte.



Na stole leží $2n$ mincí s hodnotami $1, 2, \dots, 2n$ norských korun. Číslo n je liché, a proto je i součet hodnot všech mincí lichý. Hra probíhá na střídačku. Na začátku Norbert vybere jednu minci ze stolu a Olaf rozhodne, komu připadne. V dalším kole vybere minci Olaf a Norbert rozhodne, zda ji dostane on, nebo Olaf. Poté se role opět prohodí.

Hra končí ve chvíli, kdy jeden z hráčů získá n mincí. Všechny mince, které ještě zůstaly na stole, pak automaticky dostane druhý hráč, takže oba hráči mají právě n mincí. Vítězí ten, kdo má vyšší součet hodnot svých mincí. Sára hraje za Olafa a jejím úkolem je vyhrát.

Protože je hra interaktivní, tak Norbertovy tahy určuje online rozhodčí. S rozhodčím budete komunikovat pomocí jednoduchého HTTP API. Nemusíte se bát: klienta jsme pro vás připravili a od vás se očekává jen vymyšlená strategie, jak porazit Norberta. Popis vstupu a API (níže) uvádíme zejména pro úplnost a pro případ, že si chcete napsat klienta vlastního.

Klient v Pythonu:

<https://ksp.mff.cuni.cz/viz/39-Z2-2vstup.py>

Formát vstupu: Na prvním řádku je liché kladné číslo n . Na druhém řádku je URL herního API. Komunikujte s ním pomocí HTTP dotazů GET a v každém dotazu přidejte parametr q . Hodnota $q=new$ zahájí novou partii.

Rozhodčí si stav partie předává v URL, kterou vrací v druhém řádku každé odpovědi. Při dalším dotazu použijte tuto URL a přidejte k ní parametr q s vaším požadavkem.

Formát komunikace s API:

Partii zahájíte dotazem s $q=new$. Následně se s rozhodčím střídáte v tazích. Jeden dvojtah vypadá následovně.

- Norbertova odpověď ve formátu v,m , kde v je označení vlastníka mince, kterou jste vybrali v předchozím tahu (buď 0, nebo N), a novou minci hodnoty m , o které máte rozhodnout. Jelikož Norbert začíná, tak v prvním tahu jako v vypíše X.
- Vaše odpověď, též ve formátu v,m : kdo se stane vlastníkem Norbertem vybrané mince, a hodnota mince, o které musí Norbert v příštím tahu rozhodnout. API následně vrátí odpověď v,m a cyklus se opakuje.
- Po skončení hry API vrátí WIN Olaf x nebo WIN Norbert x. Tento řádek pak odevzdáte do odevzdávacího systému. Body lze získat pouze za výsledek WIN Olaf x.

Níže je částečná ukázka hry mezi Norbertem a Olafem. Řádky označené pomocí $>$ jsme poslali my, řádky označené pomocí $<$ jsme dostali od API.

```
> q=new # začínáme novou hru
< X,25 # Norbert vybral minci s hodnotou 25
> q=N,12 # necháme mu ji, vybíráme minci 12
< 0,13 # Norbert nám ji nechává, vybírá 13
> q=0,3 # necháme si ji a vybíráme minci 3
...
WIN Olaf 12112381 # Odevzdat do odevzdávátka
```

V případě jakýchkoliv problémů napište na náš Discord nebo na e-mail zdrojaky@ksp.mff.cuni.cz a my vám rádi pomůžeme. Také vám můžeme dodat klienta v jiném jazyce, ale musíte nám napsat.

39-Z2-3 Krevní příbuzní 12 bodů

Norská královská rodina plánuje velkou oslavu a, jak už to v aristokracii bývá, musí vymyslet zasedací pořádek. Platí pravidlo, že každý člen rodiny musí sedět vedle příbuzného, který patří do širší královské rodiny, ale ne do jeho přímé linie. Královská linie je však obrovská a sestavit zasedací pořádek je náročné. Pýcha jim ale nedovolí požádat někoho o pomoc přímo, a tak vám zadají jiný úkol a doufají, že jeho vyřešení pomůže i jim.

Dostanete rodokmen zapsaný jako seznam rodičů. Sledujeme pouze královskou linii, takže každý člen rodiny kromě krále (vrchol 0) má právě jednoho rodiče a každý člen má nejvýše dva potomky. Rodokmen tedy tvoří zakořeněný binární strom. Následně dostanete velký počet dotazů. Každý

dotaz tvoří dvojice členů rodiny A a B , z nichž je vždy jeden předkem druhého a $A \neq B$.

Označme jako P toho člena z dvojice A, B , který je předkem druhého, a druhého z nich označme D . Pro každý dotaz vypište počet členů rodiny C různých od P i D , pro které je P nejnižším společným předkem dvojice (D, C) .

Nejnižší společný předek (lowest common ancestor, LCA) dvojice (u, v) je nejnižší vrchol stromu, který je předkem obou vrcholů u i v . Tohle znamená, že pokud je u předkem v , pak je LCA dvojice (u, v) právě vrchol u , ale protože hledáme vrcholy různé od A i B , tak se do výsledného součtu nepočítá.

Toto je praktická open-data úloha. V odevzdávacím systému si necháte vygenerovat vstupy a odevzdáte příslušné výstupy. Záleží jen na vás, jak výstupy vyrobíte.

Formát vstupu: Na prvním řádku je číslo N , počet vrcholů stromu. Vrcholy jsou očíslovány od 0 do $N-1$ a kořenem je vrchol 0. Na druhém řádku je N čísel, kde i -té z nich udává rodiče vrcholu i . Výjimkou je kořen – ten rodiče nemá, takže na jeho místě je -1 . Na třetím řádku je číslo Q , počet dotazů. Následuje Q řádků, každý s dvojicí čísel A a B . Zaručujeme, že $A \neq B$ a že je vždy jeden z vrcholů předkem druhého.

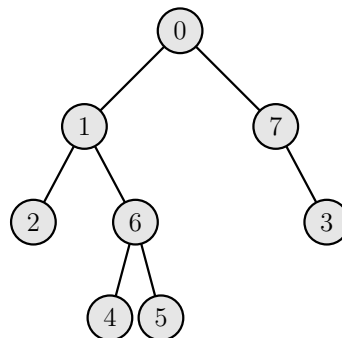
Formát výstupu: Vypište Q řádků. Na i -tém řádku bude odpověď na i -tý dotaz, tedy počet vrcholů C různých od A i B , pro které je předek z dvojice A, B nejnižším společným předkem dvojice (potomek z dvojice, C).

Ukázkový vstup:

```
8
-1 0 1 7 6 6 1 0
3
1 0
6 1
7 3
```

Ukázkový výstup:

```
2
1
0
```



Vysvětlení: V prvním dotazu platí $A = 1, B = 0$. Vrchol 0 je předek 1. Zajímá nás tedy, kolik existuje vrcholů, jejichž nejnižší společný předek s 1 je právě 0. Dle zadání se A ani B do počtu nepočítá. Pro každý zbylý vrchol si můžeme určit jeho společného předka s vrcholem 1: pro 2, 4, 5 a 6 je to 1 a pro 3 a 7 je to 0. Vyhovují tedy vrcholy 3 a 7 a výsledek je 2.

Ve druhém dotazu platí $A = 6, B = 1$. Vrchol 1 je předek 6. Pro každý vrchol si určíme jeho společného předka s vrcholem 6: pro 0, 3 a 7 je to 0, pro 1 a 2 je to 1 a pro 4, 5 a 6 je to 6. Nepočítáme A ani B , takže vyhovuje pouze vrchol 2 a výsledek je 1.

V posledním dotazu platí $A = 7, B = 3$. Vrchol 7 je předek 3. Pro každý vrchol si určíme jeho společného předka s vrcholem 7: pro 0, 1, 2, 4, 5 a 6 je to 0 a pro 3 a 7 je to 7. Opět nepočítáme A ani B , žádný další vrchol nezbývá, tudíž výsledek je 0.

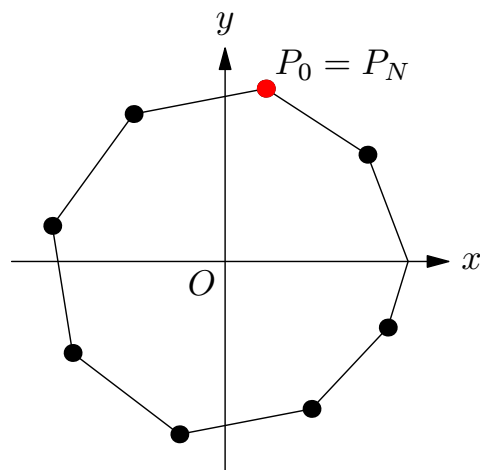
Po všech dobrodružstvích v Oslu se Kevin a Sára rozhodli vyrazit na sever podívat se na fjordy. Vlakem se tam ale pohodlně nedostanou, proto si chtějí půjčit auto. V půjčovně je však čekalo nemilé překvapení: v Norsku se smí za volant posadit jen ten, kdo zvládne zkoušku z jízdy na ledu. A tak se Kevin ocitl na zamrzlém cvičišti norské autoškoly.

Úkol zní jednoduše: udělat s autem na ledu otočku o 360 stupňů. Instruktor ale nehodnotí jen to, jestli se Kevin nepropadne tenkým ledem. Dráha, kterou auto při otočce opíše, musí uzavírat plochu menší než předepsaný počet metrů čtverečních.

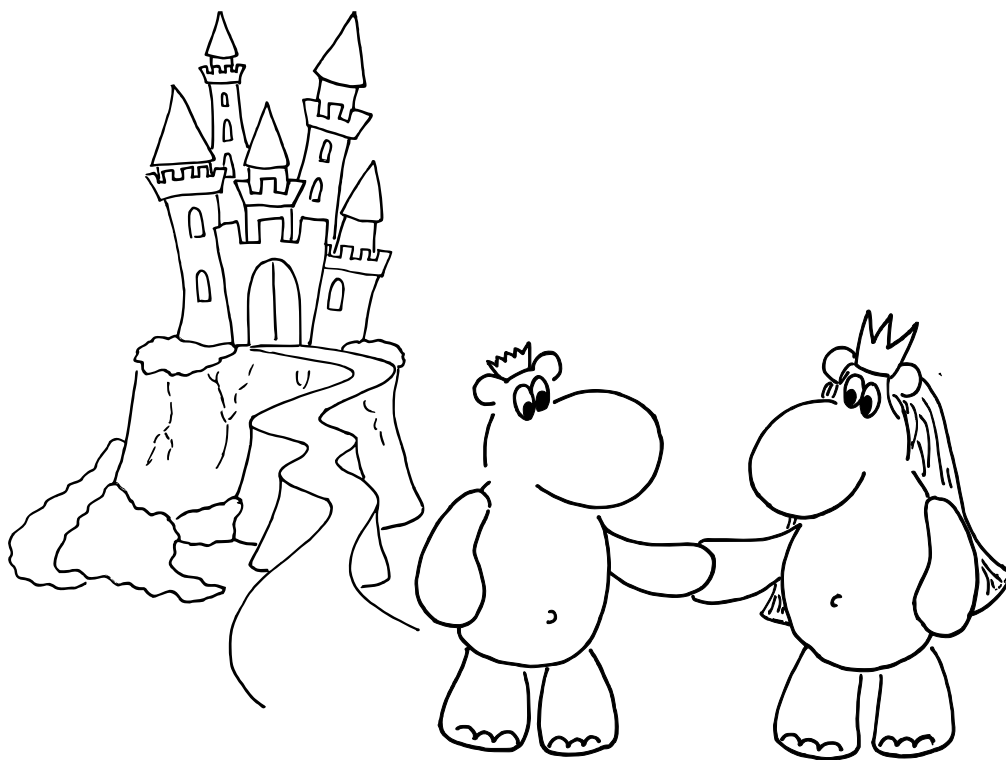
Kevin si naštěstí opatřil zařízení, které během jízdy zaznamenává souřadnice bodů dráhy. Jenže z hromady čísel na displeji sám nepozná, jestli se do limitu vešel, nebo ne. Pomůžete mu spočítat, jak velkou plochu jeho dráha uzavírá?

Na vstupu dostanete N bodů v rovině, které jsou zadány souřadnicemi (x_i, y_i) . Body jsou vrcholy *konvexního* mnohoúhelníku v pořadí, v jakém leží na jeho obvodu, a to ve směru hodinových ručiček. Vaším úkolem je navrhnout algoritmus, který spočítá obsah tohoto mnohoúhelníku. Nezapomeňte zdůvodnit, proč váš algoritmus funguje.

Mnohoúhelník je konvexní, pokud platí následující: pokud mezi libovolnými dvěma body natáhneme úsečku, nikde nám nevyleze z mnohoúhelníku ven. Při řešení této úlohy se vám může hodit přechít si začátek naší geometrické kuchařky.¹



Toto je teoretická úloha. Není nutné ji programovat, odevzdává se pouze slovní popis algoritmu. Více informací zde: <https://ksp.mff.cuni.cz/viz/tinfo>



¹ <https://ksp.mff.cuni.cz/viz/kucharky/geometrie>