

Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

36. ročník



KSP-Z

Červenec 2023

Právě se díváte na leták první série 36. ročníku KSP-Z, neboli Korespondenčního Semináře z Programování, Začátečnické kategorie. Zapojit se může **každý středoškolák i základoškolák**.

Letos bude v KSP-Z **pět sérií po čtyřech úlohách** za celkem **220 bodů**. Pokud budete mít jakoukoliv otázku, neváhejte se zeptat. Kontaktní adresy najdete v patičce na konci letáku. Přejeme hodně štěstí!

Termín série: neděle 22. října ve 32:00 (tedy další ráno v 8:00), praktické úlohy za třetinu bodů až do 29. října

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/odevzdatko/>



Zadání první série začátečnické kategorie 36. ročníku KSP

36-Z1-1 Nejčastější příjmení 8 bodů

Pokojnou atmosféru náhle protrhne záblesk světla. Maskované postavy slaňují dírami ve stropě, skáčou dovnitř okny, zní akční houslová hudba. Muž v obleku ušitým na míru se vrhne za barový pult a vytasí zbraň. Následuje efektní akční scéna, po které se zbývající záporáci dají na útěk. Hrdina si poupraví oblek, pomůže vstát dámě u baru, upije ze skleničky, která jako zázrakem zůstala stát na svém místě. Podívá se dámě do očí, a než zmizí pronásledovat padouchy, řekne jen:

„Jmenuji se Hroch. James Hroch.“



Pro Kevina je tohle oblíbená scéna celého filmu, neboť se taky jmenuje Hroch.

Stejně jako hodně jeho spolužáků a kamarádů. A taky řešitelů KSP. A obyvatel Hrochovic, kde žije. Možná je Hroch nejčastější příjmení, které se v Česku vyskytuje. A nebo je jeho statistický vzorek značně neobjektivní. . .

Aby si svoji myšlenku ověřil, začal hned stahovat z internetu nejrůznější seznamy příjmení, třeba z výsledkovek seminářů, které řešil, nebo jen znal od kamarádů.

Vyextrahovat z nich příjmení dokázal poměrně snadno. Narazil však na jiný problém: Výsledkovky řadí řešitele podle bodů. A proto příjmení, která získal, nebyla seřazena abecedně. Stejná příjmení tak byla roztroušená po celém seznamu a počítat je bylo moc obtížné.

Obrátil se tedy na vás s prosbou, jestli mu pomůžete najít v daném seznamu nejčastější příjmení. Také by ho zajímalo, kolikrát se tam vlastně toto příjmení vyskytuje.

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 zadané číslo N – velikost seznamu. Následuje N řádků, na každém z nich jedno příjmení. Příjmení sestávají jenom ze znaků anglické abecedy.

Formát výstupu: Na výstup vypíšete jeden řádek obsahující nejčastější příjmení a počet jeho výskytů, oddělené mezerou. Slibujeme, že řešení je vždy jednoznačné. Mužské a ženské verze příjmení považujeme za rozdílné.

Ukázkový vstup:

7
Novak
Hroch
Novak
Kos
Vrabec
Hroch
Hroch

Ukázkový výstup:

Hroch 3

36-Z1-2 Buldozer s krabicí 10 bodů

Kevinovi rodiče si pořád stěžují na jeho nepořádek, který se postupně rozrůstá i za hranice jeho pokoje. Kevin se proto jednoho dne rozhodl, že se s tím pokusí něco udělat. Pořídil si tedy krabici a všechny svoje věci, které jeho okolí nazývalo nepořádkem, přemístil do krabice. Pak ale zjistil, že krabice je moc těžká. Nejenže ji neunes, dokonce ji ani nezvládne tlačit po podlaze. Protože znovu přeskládat věci by dalo moc práce, rozhodl se využít pravidla „Když to nejde silou, půjde to větší silou.“ A pořídil si automatizovaný buldozer, kterým plánuje přepravovat svoji krabici.

Pro potřeby naší úlohy si můžeme představovat Kevinův pokoj jako čtvercovou mřížku. Na začátku je na nějakém políčku buldozer a na jiném krabice. Buldozer pak od Kevina dostává instrukce, které mu říkají, že se má přesunout na jedno ze sousedních políček mřížky. Pokud je na cílovém políčku krabice, tak buldozer bude tlačit do krabice a kromě přesunu buldozeru o jedno políčko se ve stejném směru posune i krabice.

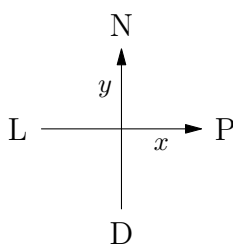
Kevin rád s buldozerem jezdí opravdu daleko, takže po chvíli na něj přestal vidět. Kevin si ovšem poctivě zapisuje, jaké instrukce postupně buldozer vykonával. Zajímalo by ho, na jaké pozici po provedení těchto instrukcí skončil buldozer a na jaké krabice. Váš úkol bude toto spočítat.

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 vstupu jsou souřadnice zadány jako dvě celá čísla oddělená mezerou:

- první z nich udává vzdálenost ve směru vlevo-vpravo (vpravo je kladný směr),
- druhé číslo udává vzdálenost ve směru dolů-nahoru (nahoru je kladný směr).

Na prvním řádku jsou čísla x_B y_B – souřadnice buldozeru. Na druhém řádku x_K y_K – souřadnice krabice. Na třetím řádku posloupnost znaků NPLD, což jsou pohyby buldozeru (Nahoru, doPrava, doLeva a Dolů).¹



Kevinův pokoj je opravdu veliký. Rozprostírá se v obou směrech mezi souřadnicemi $-10^9, \dots, 10^9$ (včetně). Máte zaručeno, že ani krabice ani buldozer nikdy neopustí pokoj (a ani se o to nepokusí).

Formát výstupu: Na prvním řádku má být x'_B y'_B – finální souřadnice buldozeru po vykonání všech instrukcí. Na druhém řádku pak x'_K y'_K – finální souřadnice krabice po vykonání instrukcí.

Ukázkový vstup:

```
1 3
2 2
PDDL
```

Ukázkový výstup:

```
1 1
2 0
```

36-Z1-3 Vejce 12 bodů

Kevin se rozhodl, že se stane farmářem. A to ne jen tak ledajakým, chce mít co nejvíce slepic v celém širém okolí.

Dnes si koupil na tržišti slepičí vejce, ze kterého se mu za K dní vylíhne jeho první slepice. Ta snese své první vejce po P dnech od narození a každé další opět po P dnech.

Z těchto vajec se vylíhnou další slepice, opět po K dnech ode dne snesení vejce, a tyto slepice budou snášet další vejce. Takto se bude Kevinova farma postupně rozrůstat.

Kevin by teď zajímalo, kolik bude mít slepic po D dnech, aby se mohl pochlubit kamarádům.

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 vstupu dostanete jediný řádek obsahující tři přirozená čísla K , P a D . K značí dobu, za jakou

se z vejce vylíhne slepice, P periodu snášení vajec danou slepicí (tedy první vejce slepice snese P -tý den svého života, další $2P$ -tý a tak stále dále – slepice neumírají). D značí počet dní, po kterých chce Kevin vědět, kolik bude mít slepic.

Slibujeme, že K , P , D nepřesáhnou $5 \cdot 10^5$. Navíc počet slepic v chovu bude po celou uvažovanou dobu nejvýše 10^{18} , proto použijte 64-bitovou proměnnou (`long long` v C++). V Pythonu se o velikost čísel nemusíte starat. Nějaké body však lze získat i za řešení, co takto velké vstupy nevyřeší.

Formát výstupu: Na jediný řádek vypíšete počet vylíhnutých slepic po D dnech. Sčítání slepic se provádí večer, tedy pokud se slepice narodí D -tý den, je třeba ji do odpovědi započítat. První den je den následující po koupení prvního vejce.

Ukázkový vstup:

```
10 4 25
```

Ukázkový výstup:

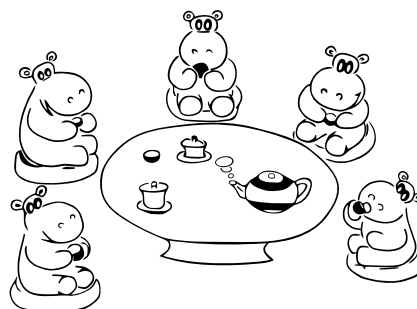
```
2
```

Po 10 dnech bude mít první slepici. Po 14 dnech bude mít stále jednu slepici, navíc ale bude mít i další vejce. Po 22 dnech již sice bude mít 3 vejce, ale pořád se mu žádné nevylíhne. Dvě slepice tedy bude mít až po 24 dnech. Následující 25. den simulace se již nic zajímavého nestane.

36-Z1-4 Čaj 14 bodů

Byl-nebyl v skrytých zákoutích pražských uliček schovaný jeden maličký obchůdek s čajem. Říkávalo se o něm různé: podle některých šlo o obyčejnou pohádku, mýtus, na mapách přece žádný obchůdek nebyl a mnoho lovců čaje se ho marně pokoušelo najít už celá léta. Jiní však tvrdili, že obchůdek existuje, ale zjevuje se jenom tomu, jehož srdce netíží žádné zlo, jen upřímná touha po dobrém čaji, a jehož kroky vedou spíše zbloudilé myšlenky než nutnost dojít do nějakého cíle.

A tak se jednoho dne tenhle obchůdek zničehonic zjevil jednomu nejmenovanému organizátorovi KSP, když se s notebookem v ruce toulal uličkami a přemýšlel nad implementací generátoru k úloze do nové série. Jelikož jeho srdce toužilo po čaji, rozhodl se záhadný obchůdek navštívit.



V obchůdku měli nabídku sestávající z mnoha pytlíků čaje visících v řadě na provázku. Některé z pytlíků přišly našemu orgovi zajímavější než jiné. Proto se rozhodl, že koupí takové pytlíky, aby jejich celková zajímavost byla co největší. Zároveň jejich celková hmotnost nesmí být příliš vysoká, jinak je všechny neunes.

Narazil však na problém – čaj může nést jenom v jedné ruce, protože ve druhé potřebuje pořád nést notebook. Z toho důvodu musí požádat o jeden souvislý úsek provázku s pytlíky.

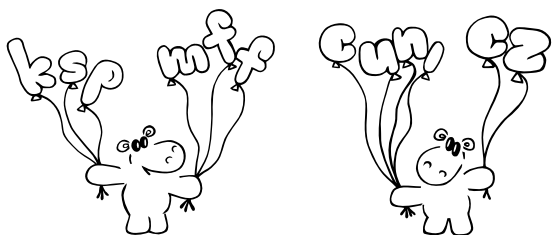
¹ Kevinův pokoj ve skutečnosti nemá podlahu sklopenou směrem nahoru, ovšem pro potřeby naší úlohy si můžeme představit pláněk pokoje nakreslený na papíře.

Pomozte mu vybrat, který úsek provázku koupí!

Algoritmus dostane na vstupu seznam pytlíků v pořadí, v jakém visí na provázku. Pro každý pytlík známe jeho hmotnost H_i a zajímavost Z_i , obě celá nezáporná čísla. Dále víme číslo M – maximální hmotnost čaje, kterou dokáže org odnést. Výstupem mají být indexy prvního a posledního pytlíku, které může vzít s sebou. Hmotnost pytlíků na daném intervalu včetně prvního a posledního nesmí nepřekročit M a zároveň součet zajímavostí musí být největší možný.

⤴ **Lehčí varianta (za 9 bodů):** Všechny pytlíky mají stejnou hmotnost. Lišit se tedy mohou jen v zajímavostech.

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



Praktický kurz programování

☰ Pokud Tě lákají praktické úlohy, ale ještě neumíš žádný programovací jazyk, můžeš se podívat na náš Základní kurz programování, kde se můžeš naučit základy Pythonu: <https://ksp.mff.cuni.cz/kurz/>.

Zdrojáky praktických úloh

☰ Řešení praktických úloh může být ze začátku složité. Velmi často i nějaká triviální technická chyba ve zdrojovém kódu programu může znamenat, že program vrací špatný výsledek – a některé chyby se ze začátku špatně hledají. Proto Ti nabízíme možnost poslat zdrojový kód programu nějaké úlohy na adresu zdrojaky@ksp.mff.cuni.cz, kde se Ti pokusíme poradit. Do emailu prosím připiš:

- Jakou úlohu by měl program řešit.
- Slovní popis, co by měl program podle Tebe dělat.

Před termínem série Ti nemůžeme radit s algoritmem, ale pomůžeme s odladěním zdrojáku. Po termínu série pak můžeme poradit i s návrhem algoritmu – získáš tak znalosti do dalších sérií.