

# Korespondenční Seminář z Programování

## ZAČÁTEČNICKÁ KATEGORIE

37. ročník



KSP-Z

říjen 2024

Právě se díváte na leták druhé série 37. 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 8. prosince ve 32:00** (tedy další ráno v 8:00), praktické úlohy za třetinu bodů až do 15. 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/odevzdatko/>



### Zadání druhé série začátečnické kategorie 37. ročníku KSP

#### 37-Z2-1 Tiskárna na kečup 8 bodů

Při čtení knih si Sára na Hroší jezero tak navykla, že hledá další způsoby, jak u něj trávit čas. „Už to mám, zřídím si na pláži stánek s tousty,“ napadlo ji. Kamarádům se záměr líbí a Kevin už demontuje svou starou 3D tiskárnu, aby ji naučil na křupavoučké plátky chleba malovat kečupem veselé obrázky. Na Zuzku tak zbyla logistika celého podniku a má z ní teď těžkou hlavu: V lednici je velká nádoba kečupu a někdo musí příliš náročné zákazníky odmítnout již při objednávce a předejít trapné situaci, když suroviny nevyzbydou. Pomozte Zuzce zachovat čest jejich stánku:

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 se nachází čísla  $T$  – počet objednaných toustů a  $K$  – množství kečupu v lednici v litrech. Následuje  $T$  řádků; na každém celé číslo udávající počet mililitrů, které se na daný toust spotřebují.

*Formát výstupu:*

Vypište čísla toustů, které přátelé vydají (číslujeme od 1). Dodržujeme striktně pořadí ve frontě – kdo může být obslužen, je obslužen, na koho už kečup nezbývá, je přeskočen.

*Ukázkový vstup:*

5 1  
3000  
250  
700  
100  
25

*Ukázkový výstup:*

2  
3  
5

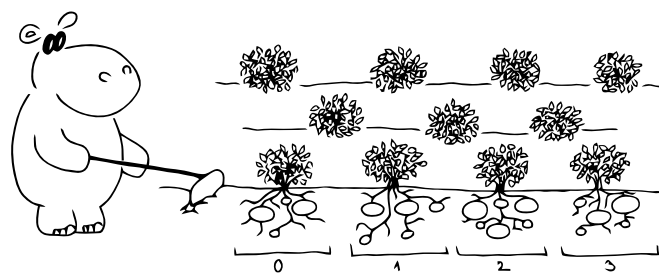


#### 37-Z2-2 Sekání obilí 10 bodů

Byl to tehdy klidný den. Vesničtí obchodníci měli dobré kurzy, ovečkám narůstala rychle vlna, ale co je nejdůležitější – obilí rychle rostlo. To se takhle Kevin procházel po svém hrochostičkovém světě, když ho z nenadání chytla ta nejhroší nemoc ze všech – *hlad!*

Pelášil na své dlouhé pole s obilím, které mu tam rostlo už 20 hodin. Jenže až tam zjistil, že jeho magickým svítkem pozeňnané motýčce zbývá už jenom jedno použití! Naštěstí měla nejnovější svítek od Karla SvítkuProdala, díky kterému dokázala motýčka posekat hned několik sousedních políček naráz.

Kevin by rád využil svou suprovou motýčku na maximum, proto s ní chce posekat co nejvíce obilí, co to jde. Pomůžete mu zjistit, kde má svou motýčku použít?



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 vstupu dostanete aktuální herní čas v hodinách  $T$ . Na druhém řádku dostanete číslo  $M$  – počet políček vedle sebe, které dokáže Kevinova motýčka posekat. Na třetím řádku dostanete velikost Kevinova pole  $P$ . Na čtvrtém řádku dostanete  $P$  čísel, kde  $P_i$  je čas, kdy bylo zasazeno obilí na  $i$ -tém políčku. Pozor, políčka indexujeme od nuly! Zaručujeme, že každé políčko bylo zasazeno před aktuálním časem. Každé zasazené obilí začíná na výšce 0 kostiček a za 1 hodinu vyroste o 1 kostičku. Po posekání získá Kevin tolik kostiček obilí, jak dlouho obilí rostlo.

*Formát výstupu:* Najděte souvislý úsek  $M$  políček, který má Kevin posekat, aby získal co nejvíce obilí. Vypište dvě čísla: index prvního políčka tohoto úseku a množství obilí v kostičkách, které takto Kevin získá. Pokud je takhle výhodných úseků více, vypište první z nich.

*Ukázkový vstup:*

4  
3  
7  
0 1 3 0 1 2 2

*Ukázkový výstup:*

3 9

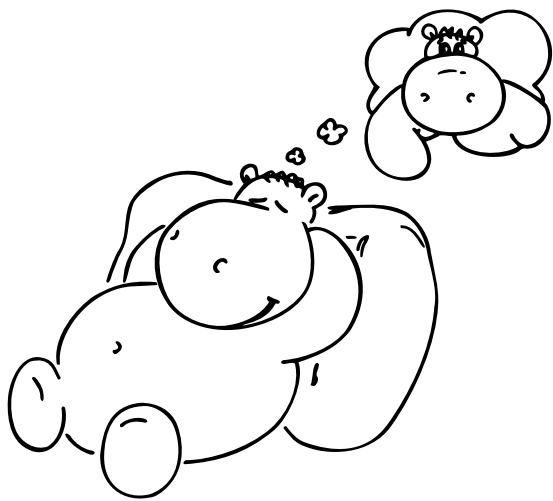
Nejvýhodnější je posekat políčka na indexech 3, 4 a 5, čímž Kevin získá  $4 + 3 + 2 = 9$  kostiček obilí.

### 37-Z2-3 Hodní orgové 12 bodů

Uběhl další měsíc a orgové se sešli na tvorbu další série. Aby jim tvorba úložek šla rychleji, tak samozřejmě začali pít čaj. Nebyl to však ledajaký čaj, ale čaj, který jim přišel z Hrocholandských polí. Tento čaj byl požehnan hrošími bohy a dával orgům možnost předvídat budoucnost.

Orgům se začaly zjevovat vize. Vize o tom, které úlohy který účastník vyřeší a které ne. A tak se rozhodli, že svoje poznatky spojí a vytvoří takovou sérii, aby co nejvíce účastníků dosáhlo na titul úspěšného řešitele.

Orgové už se rozhodli, že v sérii budou úlohy za 8, 10, 12 a 14 bodů, ale ještě neví, která úloha bude za kolik bodů. Pomůžete orgům najít nejlepší bodování úloh?



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  $U$ ,  $R$  a  $B$ .  $U$  je počet úloh,  $R$  je počet účastníků a  $B$  je hranice bodů. Na druhém řádku dostanete  $U$  čísel udávajících počty bodů úloh v neurčeném pořadí. Následuje  $R$  řádků, kde každý z nich popisuje jednoho účastníka. Na každém z nich  $U$  čísel, kde  $j$ -té číslo může být 0 nebo 1. Pokud je 1, tak účastník vyřešil  $j$ -tou úlohu, jinak 0.

Snažíme se najít obodování, při kterém co nejvíce účastníků dosáhne alespoň  $B$  bodů.

*Formát výstupu:* Na první řádek vypište počet úspěšných řešitelů pro vaše nejlepší obodování. Na druhý řádek vypište  $U$  čísel, kde  $i$ -té číslo je počet bodů, které dostane  $i$ -tá úloha.

*Ukázkový vstup:*

5 2 27  
8 8 10 12 14  
1 1 1 1 0  
1 1 1 0 0

*Ukázkový výstup:*

2  
8 8 12 10 14

Nejlepší obodování je 8 8 12 10 14 tedy všichni účastníci se stali úspěšnými řešiteli. Všimněme si že i kdybychom prohodili úlohy za 12 a 14 bodů, tak by to taky byl správný výsledek.

### 37-Z2-4 Udatná želva 14 bodů

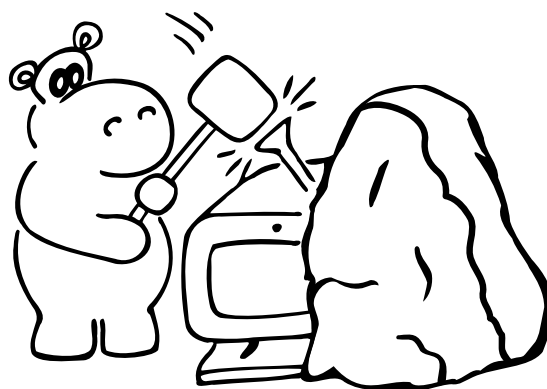
Kevin se jednoho krásného dne vydal se svou věrnou želvou do krajiny plné spadáných listů, tlejícího spadalého ovoce a sychravého počasí. Šli oba dál a dál, hnáni touhou poznat z té krásné krajiny co nejvíce. A hle, co to nevidí?

Na obzoru se rýsovala velká skála a v ní malá skulinka, sice malá, ale dost velká na to, aby se jí Kevin dokázal protáhnout. A to také udělal.

Na druhé straně ho čekalo krásné, nevidané, ba i neslychané. Stalaktity, stalagnity, stalagnáty, stalahnáty i stalamáty se blížily ze všech stran obří jeskyně. I želva se lehce pousmála nad tou nebetyčnou krásou. V tom však „BUM! PRÁSK! BŘINK!“ Kameny začaly padat ze stropu.

Ani se nenadál a stál mezi hromadami kamení. Naštěstí měl s sebou stále svou želvu s nezničitelným bombuvzdorným krunýřem. Ta se byla schopna mezi kameny jistě protáhnout. Věděl, že když odpálí hodně hromad kamenů, tak se možná mezi hromadami protáhne a stihne doběhnout ven dřív, než se celá jeskyně sesype. Měl však jen jednu bombu a svou hrdinnou želvu.

Pomozte mu vymyslet, kam má Kevin poslat svou udatnou želvu, aby dokázal utéct.




Na vstupu dostanete mřížku tvořenou z prázdných polí a kamenů, která představuje mapu jeskyně. Kameny jsou buď zničitelné, nebo nezničitelné. Na libovolné pole (i s kamenem) můžete odnavigovat želvu, která tam pak i vybuchne. Bomba šíří svůj výbuch přes prázdná pole do všech 4 směrů v přímé čáře a ničí všechny zničitelné kameny po cestě. Když narazí na nezničitelný kámen, tak se šířit přestane. Najděte pole, odkud bomba zničí nejvíc kamenů. Ale pozor, času zbývá málo. Výpočet tedy musíte stihnout v čase  $\mathcal{O}(N)$ , kde  $N$  je velikost vstupu.

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 zdro-

jovém 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](mailto: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.

