

Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

33. ročník

KSP-Z


Únor 2021


Vítejte u čtvrté série 33. ročníku **KSP-Z**, neboli Korespondenčního Semináře z Programování, Začátečnické kategorie.

Již uběhla více než polovina z letošních pěti sérií, ale stále se může zapojit **každý středoškolák i základoškolák**. Situace bohužel neumožňuje konání jarního soustředění, ale budeme k účasti v KSP-Z přihlížet při výběru účastníků **podzimního soustředění**, které se snad již povede uspořádat. Svoje šance na přijetí můžete zvýšit účastí v této sérii KSP-Z.

Pokud budete mít jakoukoliv otázku, neváhejte se zeptat. Kontaktní adresy najdete v patičce letáku. Přejeme hodně štěstí!

Termín série: neděle 4. dubna ve 32:00 (tedy další ráno v 8:00), praktické úlohy za třetinu bodů až do 11. dubna

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/odevzdej.cgi>



Zadání čtvrté série začátečnické kategorie 33. ročníku KSP

33-Z4-1 Střílení do terče 8 bodů

Kevin má dnes dobrý den! Byl v kině, kde se díval na nový film Jamese Bonda, a řekl si, že se naučí střílet z pistole. A kde jinde by se to mohl naučit než na střelnici.

Asi vám nemusím vysvětlovat, jak vypadá terč... Jsou to soustředné kružnice různých poloměrů, jednotlivá mezikružší jsou různě obodovaná. Jenže když si Kevin poprvé zastřílel, tak zjistil, že z něho agent s povolením zabíjet ještě dlouho nebude – kulky létaly do všech směrů. Proto si úkol zjednodušil a dal si za cíl terč vůbec trefit. Co říkáte, dobrý začátek, ne?

Co se ale Kevinovi musí nechat, je kadence výstřelů. Protože střílí jako kulomet, je děsná otrava vyhodnocovat, kolik kulek trefilo terč... Naštěstí Kevin ví, kam dopadly jednotlivé kulky, takže jen stačí zjistit, jestli trefily terč či ne.

Formát vstupu: Na prvním řádku vstupu dostanete tři desetinná čísla X , Y a R udávající pozici středu terče v rovině a poloměr terče. Na dalším řádku bude číslo N udávající počet kulek, které Kevin vystřelil. Na dalších N řádcích následují pozice dopadu jednotlivých kulek (x -ová a y -ová souřadnice).

Formát výstupu: Nechť K je počet kulek, které trefily terč. Pak na K řádků vypište čísla jednotlivých kulek, které trefily cíl. Kulky číslujeme od jedné. Čísla vypisujte vzestupně (tak, jak jsou na vstupu).

Ukázkový vstup:

1 1 1.5
3
0 1
0.8 -1
2 2

Ukázkový výstup:

1
3

Nemusíte se bát zaokrouhlovacích chyb. Generujeme vstupy, u kterých to nemusíte řešit.

33-Z4-2 Hesla nebo odpad? 10 bodů

Sáře se povedlo najít na internetu soubor, který o sobě tvrdil, že obsahuje spoustu hesel do školního známkovacího systému. Než se tím pochlubí Kevinovi a Petrovi, tak se

rozhodla soubor (z čistě vědeckého zájmu) prozkoumat. Po chvíli zjistila, že některá hesla sice fungují, ale většina jich je asi falešná: „Tohle přeci ani nemůže být správné heslo, vždyť to nemá velké písmeno na třetím místě a číslici na posledním,“ povzddechla si.

Školní systém má totiž docela striktní pravidla na to, jaká hesla se v něm dají nastavit.¹ Heslo se smí skládat z číslic a z malých i velkých písmen anglické abecedy, nic jiného se v něm vyskytovat nemůže. Může být libovolně dlouhé, ale je stanoveno, čím musí začínat a čím končit.

Správce systému stanoví *předpis* hesla, který se skládá ze znaků a , A , 0 a $*$. Znak a zastupuje jedno libovolné malé písmeno, znak A zastupuje jedno libovolné velké písmeno, znak 0 zastupuje jednu libovolnou číslici a znak $*$ zastupuje jakkoliv dlouhou posloupnost libovolných znaků (tedy i prázdnou posloupnost). V předpisu se znak $*$ vyskytuje právě jednou. Raději upozorňujeme, že spřežku ch považujeme za dvě písmena c a h .

Heslo vyhovuje předpisu tehdy, když na odpovídajících pozicích v předpisu jsou znaky, které zastupují dané znaky hesla. Například předpis $Aaa*00A$ říká, že heslo musí začínat velkým písmenem následovaným dvěma malými písmeny a musí končit dvěma číslicemi a velkým písmenem, uprostřed hesla může být libovolný řetězec. Heslo $Hrouda33X$ tomuto předpisu odpovídá (za hvězdičku se „schová“ uda), stejně tak $Max12Z$ (hvězdička je prázdná), ale třeba $MaPa12C$ už ne (nesedí třetí pozice, kde by mělo být malé písmeno).

Sára by chtěla vědět, která hesla ze souboru mohou být správná a která ani nemá smysl zkoušet.

Formát vstupu: Na prvním řádku vstupu dostanete *předpis* skládající se ze znaků $aA0*$ ($*$ v něm bude právě jednou). Na druhém řádku pak následuje číslo N udávající počet hesel ke zkontrolování. Na dalších N řádcích pak dostanete hesla skládající se z číslic a malých a velkých písmen anglické abecedy.

Formát výstupu: Pro každé heslo ze vstupu vypište na samostatný řádek **ano**, pokud heslo odpovídá předpisu, nebo **ne**, pokud tomu tak není (celkem tedy vypište N řádků).

¹ Což je mimochodem dost špatná praxe.

Ukázkový vstup:

Aaa0*0a00
5
Map7z9x24
Ded55555k11
muj4zub5v55
Guh5a12
ctra74axx

Ukázkový výstup:

ano
ano
ne
ne
ne

33-Z4-3 Kluzké bludiště 12 bodů

Kevin se Sárou si postavili robota, který umí jezdit čtyřmi směry: nahoru, dolů, doprava a doleva. Také si pro robota postavili bludiště, které ovšem bylo na příliš kluzké podlaze. Když se v bludišti robot rozjel nějakým směrem, tak jel rovně, dokud nenarazil na stěnu. Když robot stojí, dá se mu říct jeden ze čtyř směrů, kterým se má vydat dál.

Roboti jsou sice robustní, ale neustálé narážení do stěn jim nedělá dobře. Kevin by chtěl pro každé uspořádání bludiště najít takovou cestu do cíle z počáteční pozice robota, která bude obsahovat co nejméně nárazů do stěn. Cíl je prohlubeň v bludišti, tudíž stačí přes cíl přejet a nemusí se na něm robot zastavovat o stěnu. Celé bludiště je zvenčí ohraničeno stěnou, která není zobrazena v jeho nákresu, takže z něj není možné vyjet ven.

Formát vstupu: Na prvním řádku vstupu dostanete dvě čísla R a S udávající rozměry bludiště. Poté následuje R řádků po S znacích představujících bludiště. Znak S značí místo, kde robot začíná, znak C místo, kam se robot musí dostat, znak $\#$ označuje stěnu a znakem $.$ je označené volné políčko.

Formát výstupu: Na jeden řádek vypíšete příkazy pro robota popisující cestu do cíle s nejmenším počtem nárazů do stěny. K dispozici jsou čtyři možné směry z pohledu na bludiště (tedy neovlivňuje je směr, ze kterého přijel robot): Nahoru, Dolů, vPravo a vLevo.

Ukázkový vstup:

6 5
S
. # . . .
. . C . .
.
. . . .
. . .

Ukázkový výstup:

PDLNP

Nápověda: K řešení této úlohy by vám mohla pomoci naše kuchařka o grafech dostupná na našem webu.

33-Z4-4 Opačná úloha 14 bodů

Kevin si dělal doma pořádek a ve svých starých řešeních KSPčka našel jedno zajímavé. Bylo to řešení na úlohu nalezení úseku posloupnosti s daným součtem. Kevin se do svého řešení začel . . .

Zadání úlohy: Dostanete na vstupu posloupnost N celých čísel a_0, a_1, \dots, a_{N-1} a číslo K . Úkolem je najít souvislý úsek posloupnosti takový, že součet všech čísel v tomto úseku dá právě číslo K (tedy najít indexy i a j takové, že $a_i + a_{i+1} + \dots + a_j = K$). Při více takových úsecích stačí

najít libovolný z nich. Pokud takový úsek neexistuje, mělo by to řešení poznat.

Kevinovo řešení: Načteme si vstup do pole a pořídíme si na začátku dva indexy i a j nastavené na nulu a součet S nastavený na hodnotu prvního prvku pole (na indexu 0). Pak budeme ve smyčce provádět následující:

- Pokud $S < K$: Přičteme k indexu j jedničku. Pokud je $j = N$, tak jsme ho posunuli za konec pole, skončíme a oznámíme neúspěch. Jinak přičteme k S hodnotu prvku pole na indexu j .
- Pokud $S > K$: Odečteme od S hodnotu prvku pole na indexu i a poté k i přičteme jedničku.
- Pokud $S = K$: Vypíšeme indexy i a j jako nalezené řešení a skončíme.


Kevin se na své staré řešení podíval a všiml si, že neuvedl jeho časovou složitost a ani nevysvětlil, proč by mělo fungovat. Jak o něm přemýšlel, tak si ani nebyl jistý, zdali vůbec funguje správně a vrátí vždy správný výsledek.

Váš úkol v této úloze bude opačný, než je většinou zvykem. Zamyslete se nad výše popsanou úlohou a jejím řešením a zodpovězte na **následující otázky**:


1. Funguje řešení správně? Buď zdůvodněte, proč funguje a vždy najezte úsek součtu K , pokud takový existuje, nebo nalezněte protipříklad, na kterém řešení selže. Zdůvodnění formulujte co nejpečlivěji.
2. Jak rychle (s jakou časovou složitostí) toto řešení vrátí výsledek pro vstup o N číslech?

Otázky zodpovězte **tříkrát pro různé vstupy**: Vstup jen z celých kladných čísel, vstup jen z celých záporných čísel a vstup z libovolných celých čísel.

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/kurzy/zkp/>.

Zdrojky 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.



KSP pro vás připravují studenti Matematicko-fyzikální fakulty Univerzity Karlovy.

Webové stránky:
<https://ksp.mff.cuni.cz/>

E-mail:
ksp@mff.cuni.cz

Organizátoři a kontakty:
<https://ksp.mff.cuni.cz/kontakty/>