

Základy algoritmického myšlení

– sbírka příkladů v0.1

Jednodušší příklady

Balancování

Definujme si dvě kategorie znaků: písmena a mezera. Napište program, který vyzve uživatele k zadání textu a pak zkontroluje, zda je zadaný text vyvážený a výsledek své kontroly vypíše na obrazovku. Text je vyvážený, pokud je počet mezer mezi deseti až dvaceti procenty z celkového množství zadaných znaků.

Konstrukce trojúhelníku

Vytvořte program, který uživatele poprosí o zadání délky strany trojúhelníku. Poté postupně vypíše všechny možné takové kombinace celočíselných hodnot (menších než 12), které reprezentují možné délky dalších dvou stran trojúhelníku.

Připomínám, že součet délek dvou stran trojúhelníku není nikdy menší než délka strany třetí.

Detekce anomálií

Napište program, který bude opakovaně vyzývat uživatele k zadání čísla. Po každém zadání program zvaží, zda se uživatelem zadané číslo neodchyluje o průměru dosud zadaných čísel o víc, než tři. Pokud ano, pak program skončí. Pokud ne, program pokračuje ve výzvách.

Úsporný zápis

Vytvořte ve Scratchi program tak, který se zeptá uživatele na text. Poté text vypíše znovu (v jedné bublině), ale bez samohlásek. Příklad • Vstup: Ahoj Duchu. • Výstup: Ahj Dch.

Pokud si chcete příklad zkomplikovat, upravte program tak, aby vynechal pouze ty samohlásky, která jsou v zadaném textu méně než dvakrát.

Postupka

Napište program v Pythonu nebo se Scratchi, který bude opakovaně vyzývat uživatele k zadání čísla. Po každém zadání program zvaží, zda uživatelem zadané číslo není menší, než součet dosud zadaných čísel. Pokud ano, pak program skončí. Pokud ne, program pokračuje ve výzvách.

Složitější příklady

Abeceda

Vytvořte program, který se bude ptát na písmena tak dlouho, dokud uživatel nezadá prázdný řetězec. Zadaná písmena pak program vypíše seřazená podle abecedy od největšího po nejmenší. Očekávejte, že uživatel zadává pouze písmena a, b nebo c. Pokud uživatel zadá něco jiného, program skončí.

Cifry

Napište v Pythonu nebo Scratchi program, který se uživatele zeptá na číslo q. Program poté vypíše všechna tříciferná čísla jejichž ciferný součet je roven hodnotě q.

Titulkovač

Napište program, který uživatele vyzve k zadání textu. Poté text vypíše na obrazovku, ale všechna první písmena slov (žádná jiná) budou vyměněna za verzáčky (velká písmena).

Čísla a cifry

Napište program, který bude od uživatele načítat čísla tak dlouho, dokud uživatel nezadá cokoli jiného, než přirozené kladné číslo. Poté program vypíše všechna taková zadané čísla jejichž ciferný součet je liché číslo.

Šifra

Napište program, který zašifruje uživatelem zadaný text pomocí šifry "první písmeno" a zašifrovaný výsledek vypíše na obrazovku. Princip šifry "první písmeno" spočívá ve přečtení pouze počátečních písmen slov. Např. pokud uživatel zadá "heslo", výsledek může vypadat třeba: "house eye seer lemon only". Seznam všech běžných slov najdete v soboru zde: <http://goo.gl/Ju58ys> Soubor se slovy můžete nainportovat ve Scratchi do proměnné typu seznam (list).

Unikátní

Napište program ve Scratchi, který poprosí uživatele o zadání textu. Poté vypíše všechna slova, která se v textu vyskytovala právě jednou.

Scrabble

Napište program, který si od uživatele nechá zadat množinu písmen. Poté vypíše z tohoto <http://goo.gl/Ju58ys> seznamu všechna taková slova, která se dají z těchto písmen sestavit. Písmeno se ve vypsáném slově může vyskytovat maximálně tolikrát, kolikrát se je v zadané množině.

Validace data

Napište program, který bude od uživatele načte datum narození ve formátu DD-MM-YYYY a pak vypíše, zda je datum platné. Data před 01-05-1920 nebo po 08-06-2022 nejsou platná. Pokud implementujete i kontrolu přestupného roku, dostanete bod navíc. Zde jsou nějaké příklady zadání, které platné nejsou: • 3-5-2022 • 31-04-1999 • 30-02-2015 • 02-02-1920

Rýma

Vytvořte ve Scratchi nebo Pythonu program, který od uživatele načte slovo a poté vypíše až tři běžná další slova, která se na zadané slovo rýmují. Pro zjednodušení pracujte s premisou, že dvě slova se rýmují, pokud mají stejná dvě poslední písmena. Seznam všech běžných slov najdete v soboru zde: <http://goo.gl/Ju58ys> Například, pokud uživatel zadá "enrich", program může vypsát třeba: • which • ouch • sandwich

Nátěr

Vytvořte ve Scratchi nebo Pythonu program, který uživatele poprosí o zadání množství barvy, které má k dispozici. Počítejte s tím, že uživatel zadá číslo označující hodnotu v metrech čtverečních, tj. kolik metrů čtverečních se dá disponibilní barvou akorát pokrýt. Program pak vypíše všechny možné kombinace dvojic celých čísel, které označují rozměry takových stěn, která se dá barvou beze zbytku natřít. Jinými slovy -- vypíše rozměry všech stěn, jejichž plocha (obsah) je stejná jako uživatelem zadané číslo.

Palindrom

Napište v Pythonu nebo Scratchi program, který od uživatele načte text a poté vypíše, zda jde o palindrom. Mezery ignorujte.

Posun

Napište program v Pythonu nebo ve Scratchi, který nechá uživatele načíst text ve formátu číslice-slovo (např 1-ahoj). Program poté vypíše zadané slovo. Písmena však budou “posunutá v abecedě” o hodnotu zadaného čísla. Pro 1-ahoj program vypíše bipk. Posunem abecedě je myšlena substituce písmene za písmeno, které je v abecedě na pozici, která je daná hodnotou posunu. Např. písmeno a posunuto o dvě se mění na c. Nebo písmeno y posunuto o tři se mění na b. Předpokládejte, že uživatel zadá jen malá písmena bez diakritiky

Složité příklady

Rodné číslo

Napište program, který zkontroluje uživatelem zadané rodné číslo podle kritéria dělitelnosti jedenácti a vypíše výsledek: platné/neplatné.

Dělitelnost jedenácti lze snadno zkontrolovat i zpaměti: kritériem dělitelnosti jedenácti je rozdíl součtu cifer na sudých a lichých místech a jeho dělitelnost jedenácti, např. u čísla 736028/5163 je součet číslic na lichých místech $7 + 6 + 2 + 5 + 6 = 26$, součet číslic na sudých místech $3 + 0 + 8 + 1 + 3 = 15$, rozdíl $26 - 15 = 11$ je dělitelný jedenácti, takže celé rodné číslo je dělitelné jedenácti a tedy formálně platné.

https://cs.wikipedia.org/wiki/Rodn%C3%A9_%C4%8D%C3%ADslo#Kontroln%C3%AD_%C4%8D%C3%ADslice

Dodatečné pokyny: Pokud uživatel umístí někde do čísla symbol “/”, ignorujte jej (zároveň jej ale nevyžadujte). Napište program tak, aby stejně fungoval i pro čísla s vyšším počtem cifer, tj. Např pro číslo 57132421660487.

Zakázaná slova

Napište program, který od uživatele načte text, pak nahradí všechna zakázaná slova jejich korektními alternativami a pak tento text vypíše pět na obrazovku. Jak vypadá korektní alternativa zakázaného slova? První a poslední písmeno zůstává stejné, ostatní písmena se mění na hvězdičku. Např.: “Brno” → “B**o” nebo “Mikýř” → “M***ř”. Na velikosti písmen záleží.

Zakázaná slova jsou: ● kolegyňka ● Brno ● manža ● Aš ● pimpula

Program tedy např. na vstup: “Prý je tvá kolegyňka pimpula, kAšli už na ni.”, vypíše: “Prý je tvá k*****a p*****a, kAšli už na ni.”

Pokud programujete v Pythonu nesmíte použít funkci replace ani sub

Piškvorky pro jednoho

Napište program, který od uživatele bude postupně načítat taky v piškvorkách (např. A2, C4, atd...) a skládat si kameny na pomyslnou desku velikosti 9x9 polí. Pokud program odhalí neplatný tah, tj. Buď tah na místo, kde už kámen je, nebo tah mimo desku, nebo jakékoli neplatné zadání, program vypíše chybu a skončí. Pokud program odhalí tři kameny v linii vedle sebe (vertikálně nebo horizontálně), pak uživateli pogratuluje a skončí.

Bigramy

Vytvořte program, který od uživatele načte textový řetěc a poté vypíše právě všechny bigramy (dvojice znaků) v něm obsažené více než jednou. Mezeru nepočítejte jako znak.

Příklad: Na vstup: "Jejím cílem je zabezpečit a racionalizovat sociální informační a komunikační procesy." Vypadá výstup: "je, ci, ní, ač, čn, oc"