# Zadání soutěžních úloh

# Kategorie programování žáci

## duben 2025

## Soutěž v programování – 38. ročník

## Krajské kolo 2024/2025

Úlohy můžete řešit v libovolném pořadí a samozřejmě je nemusíte vyřešit všechny. Za každou úlohu můžete dostat maximálně 10 bodů, z nichž je většinou 9 bodů vyhrazeno na hodnocení funkčnosti programu, jeho shody se zadáním a efektivity a jeden bod na dokumentaci a přehlednost zdrojového kódu (vhodné členění zdrojového kódu, vhodně zvolené názvy identifikátorů, komentáře na místech, kde je to potřeba atd.). Body získané za každou úlohu se ještě násobí koeficientem, který odráží složitost úlohy.

Na řešení úloh máte 4 hodiny čistého času.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory a kam máte ukládat vaše řešení.

### Úloha 1 – Hrátky s čísly - koeficient 2

Napište program, který pro zadané vstupní číslo N <= 16 nalezne N po sobě jdoucích celých čísel, která nejsou prvočísla.

Pro každé z těchto čísel uveďte nějakého z jeho dělitelů.

**Příklad:**

N = 3

**Výstup:**

24 = 2\*12, 25 = 5\*5, 26 = 2\*13

**Příklad:**

N = 5

**Výstup:**

120 = 2\*60, 121 = 11\*11, 122 = 2\*61, 123 = 3\*41, 124 = 2\*62

### Úloha 2 – Vonáskovy řetězce - koeficient 1

Pomozte profesoru Vonáskovi s rozborem dlouhých molekulových řetězců. Zajímá ho počet stejných vazeb mezi atomy nebo submolekulami.

Pro uživatelem zadaný textový soubor zjistěte četnosti všech dvojic za sebou jdoucích znaků vyskytujících se v souboru.

Výstupem je seznam dvojic a jejich četnost.

Dvojice a četnost jsou odděleny mezerou.

Seznam je seřazen od nejvyšší četnosti.

Při stejné četnosti se dvojice řadí podle ASCII kódů aktuálního prostředí.

Při zpracování vstupu ignorujte znaky nového řádku (CR/LF).

Malé a velké znaky se rozlišují ('a' není 'A').

Vstupní a výstupní soubory z příkladů najdete v příloze (Px.txt, PxV.txt).

| **Příklad 1** |  |
| --- | --- |
| **Vstup:** | Abrakadabra hokuspokus |
| **Výstup:** | br 2 ku 2 ok 2 ra 2 us 2 h 1 Ab 1 a 1 ab 1 ad 1 ak 1 da 1 ho 1 ka 1 po 1 sp 1 |

| **Příklad 2:** |  |
| --- | --- |
| **Vstup:** | Dlouhé molekulové řetězce jsou zábavné! |
| **Výstup:** (1 řádka) | lo 2 ou 2 é 2 j 1 m 1 z 1 ř 1 Dl 1 av 1 ba 1 ce 1 e 1 ek 1 et 1 hé 1 js 1 ku 1 le 1 mo 1 né 1 ol 1 ov 1 so 1 tě 1 u 1 uh 1 ul 1 vn 1 vé 1 zc 1 zá 1 áb 1 é! 1 ěz 1 ře 1 |

| **Příklad 3** |  |
| --- | --- |
| **Vstup:**  (3 řádky) | Ignorujte znaky nového řádku (CR/LF).  Rozlišujte malé a velké znaky ('a' není 'A'). |
| **Výstup:** (1 řádka) | ( 2 n 2 z 2 ). 2 ak 2 e 2 jt 2 ky 2 na 2 no 2 te 2 uj 2 y 2 zn 2 é 2 ' 1 a 1 m 1 v 1 ř 1 ' 1 ') 1 'A 1 'a 1 (' 1 (C 1 .R 1 /L 1 A' 1 CR 1 F) 1 Ig 1 LF 1 R/ 1 Ro 1 a 1 a' 1 al 1 dk 1 el 1 en 1 gn 1 ho 1 iš 1 ku 1 ké 1 li 1 lk 1 lé 1 ma 1 ne 1 ní 1 o 1 or 1 ov 1 oz 1 ru 1 u 1 ve 1 vé 1 zl 1 šu 1 ád 1 éh 1 í 1 řá 1 |

### Úloha 3 – Vícecestná substituce (šifrování) - koeficient 2

Spolužáci si chtějí během vyučování vyměňovat tajné zprávy tak, aby jejich učitel nepoznal, o čem si píšou. Proto si vymysleli vlastní šifrovací slovník, podle kterého překládají jednotlivá písmena. Jejich učitel byl ale velmi šikovný a většinu slovníku dokázal rozluštit.

Aby ho přelstili, rozhodli se aplikovat substituci opakovaně – tím docílili toho, že písmena byla zaměněna vícekrát za sebou a jejich zprávy se staly obtížněji dešifrovatelnými. Tento způsob šifrování se jim ale brzy stal příliš pracným, a proto potřebují tvou pomoc s vytvořením programu, který dokáže text zašifrovat automaticky.

**Vytvořte program, který:**

1. **Načte substituční slovník** ze souboru slovnik.txt. Každý řádek obsahuje dvojici znaků oddělenou středníkem – první znak je původní písmeno, druhý znak je písmeno, kterým se má nahradit. Slovník obsahuje všechna písmena abecedy právě jednou a každé písmeno má právě jeden překlad.
2. **Umožní uživateli zadat text**, který chce zašifrovat.
3. **Umožní uživateli zadat počet iterací šifrování** (kolikrát se má substituce provést).
4. **Vypíše výsledný zašifrovaný text**.

**Poznámka:** Písmeno CH se ve slovníku nevyskytuje. Program tedy pracuje pouze s písmeny C a H zvlášť.

**Druhá část úlohy: Dešifrování**

Rozšiř svůj program tak, aby kromě šifrování umožňoval i dešifrování.

1. **Uživatel si zvolí režim –** šifrování nebo dešifrování.
2. **Při dešifrování se substituce aplikuje opačně**, přičemž se bere v úvahu počet iterací, který byl použit při šifrování.

**Výsledkem dešifrování by měl být původní text.**

### Úloha 4 – Ostrovy v moři - koeficient 3

Dostali jste přístup k datům z jednoho ze satelitů námořnictva. Satelit má speciální obrazový senzor, kterým detekuje, zda se pod ním nachází pevná půda, či voda. Tedy jeho výstupem je obraz, jehož pixely nabývají hodnoty 0, voda, či 1, pevná půda.

**Vaším úkolem je určit počet a rozlohu ostrovů na aktuálním snímku.**

Zároveň by nebylo od věci pro přehlednost interaktivní zobrazení.

Výsledný program tedy zobrazí **interaktivní mapu**, modrou moře a žlutou pevnou půdu. Jakmile se uživatel kurzorem nachází nad nějakým ostrovem, vedle kurzoru se zobrazí obsah jeho povrchu. Pokud se nachází nad mořem, vedle kurzoru se nic nezobrazuje.

**Zobrazovaná mapa musí obsáhnout celé okno.**

Jako jeden ostrov se počítá takový útvar, který lze obeplout loďkou. **Tedy pokud se dvě plochy dotýkají diagonálně, jedná se o jeden ostrov.** Pokud nějaký ostrov zasahuje mimo snímek, předpokládejte **všude okolo pouze vodu.**

Velikost jednoho pixelu odpovídá ve skutečnosti 100m x 100m povrchu Země. Výsledný obsah zobrazte jako desetinné číslo zaokrouhlené na dvě desetinná místa. Jako jednotky použijte kilometry čtverečné.

**Rozlišení obrazu je 420 x 240 pixelů.** (šířka x výška)

Váš program tedy musí:

1. Umožnit výběr souboru snímku
2. Zobrazit mapu dle zadání