

Sbírka příkladů

verze 1.0 – 2.1.2005

Rudolf Kryl

Sbírka má pomoci studentům k přípravě na praktický test. Student, který „umí programovat“, umí ladit a zvládne algoritmicky úlohy této sbírky by neměl mít s praktickým testem velké problémy.

Nelze ji však chápat tak, že u praktických testů student může dostat jen úlohu z této sbírky.

- 1) Vytvořte podprogram, který prověří zda zadaný znakový řetězec (složený jen ze závorek) je správně uzávorkovaný.
- 2) Varianta: Totéž s více typy závorek, které přečteme ze vstupu (mohou být i dvouznakové).
Např.: ()

[]
{ }
(* *)
/* */

- 3) Napište program, který vypíše všechny rozklady kladného celého čísla N na kladné celé sčítance. Rozklady, které se liší jen pořadím sčítanců, považujeme za totožné. Např. 5 lze vyjádřit sedmi způsoby:

$$5 = 4+1 = 3+2 = 3+1+1 = 2+2+1 = 2+1+1+1 = 1+1+1+1+1$$

Při výpisu nezáleží na pořadí jednotlivých rozkladů ani na pořadí sčítanců v nich.

- 4) Vstupem programu jsou čísla N a K.
Nalezněte všechny K-tice celých kladných čísel $x_1, x_2, x_3, \dots, x_{(K-1)}, x_K$, pro které platí
 $x_1 \leq x_2 \leq x_3 \leq \dots \leq x_{(K-1)} \leq x_K$ a $x_1 + x_2 + x_3 + \dots + x_{(K-1)} + x_K = N$.
Hodnotí se i rychlost výpočtu.

- 5) Nalezněte všechny možné výplaty dané částky N Kč pomocí platidel o hodnotách 1,2,5,10,20,50, 100 Kč. N je zadané celé kladné číslo menší než 1000. Hodnotí se i rychlost výpočtu. Pokud možno by program měl hodnoty platidel načíst z dat.
- 6) Do posloupnosti číslic 1 2 3 4 5 6 7 8 9 zapsané v tomto pořadí vložte mezi některé číslice znaménka + nebo - tak, aby byla hodnota takto vzniklého výrazu rovna danému číslu N, které je zadáno na vstupu programu. Nalezněte všechna řešení.
Např. pro N=95 je jedním z řešení výraz: 123+4-56+7+8+9 .
- 7) Všechny permutace čísel 1,2,...,N seřadíme myšlenkově do posloupnosti podle lexikografického ("slovníkového") uspořádání. Sestavte podprogram pro „přímé“ zjištění pořadového čísla zadané permutace

- 8) Je dáno nejvýše 7 celých čísel z intervalu $\langle 1, 11 \rangle$. Program je přečte z textového souboru. Vložte mezi čísla znaménka +, -, * a / tak, aby hodnota takto vzniklého výrazu byla rovna číslu X typu longint zadanému ze vstupu. Pořadí čísel ve výrazu je neměnné a je určeno jejich seřazením na vstupu. Před první z čísel se žádné znaménko nedává. Při vyhodnocování předpokládejte stejné priority operací jako jsou v Pascalu. Operátor / představuje celočíselné dělení ("div" v Pascalu). Stačí odpovědět na otázku, zda takový výraz existuje a pokud ano, nalézt jedno libovolné řešení.

Př.: pro posloupnost čísel 11, 2, 5, 8, 3, 11, 2 a $X=21$ bude řešením úlohy výraz
 $11-2*5+8/3*11-2$.

- 9) Kostka domina je tvořena dvěma poli. Každé z nich může být prázdné nebo může obsahovat jistý počet teček od 1 do 6. Kostky se skládají do řady tak, že vedle sebe stojící kostky musí sousedit stejnými poli.

Vstupem vašeho programu bude seznam kostek domina, které máte k dispozici. V seznamu se může opakovat i více stejných kostek. Zjistěte jakou nejdelší řadu lze z těchto kostek sestavit. Vytiskněte tuto řadu a její délku měřenou počtem kostek. Stačí jedno řešení.

- 10) Všechny permutace čísel 1,2,...,N seřadíme myšlenkově do posloupnosti podle lexikografického ("slovníkového") uspořádání. Sestavte podprogram pro „přímé“ nalezení permutace k zadanému pořadovému číslu

- 11) Všechny permutace čísel 1,2,...,N seřadíme myšlenkově do posloupnosti podle lexikografického ("slovníkového") uspořádání. Sestavte podprogram pro „přímé“ zjištění v pořadí následující permutace k zadané permutaci.

- 12) Necht' A, B jsou dvě kladná celá čísla. Řekneme, že číslo A je obsaženo v čísle B, jestliže dekadický zápis čísla A lze získat vynecháním žádné, jedné nebo několika cifer z dekadického zápisu čísla B. Pořadí zbývajících cifer přitom zůstane zachováno.

Napište podprogram, který k danému kladnému celému číslu N určí a vypíše všechna čísla v něm obsažená. Na pořadí jejich výpisu nezáleží, ale žádné z čísel se nesmí na výstupu opakovat.

Např. číslo $N=1223$ obsahuje čísla 1223, 223, 123, 122, 23, 22, 13, 12, 1, 2, 3.

- 13) Necht' A, B jsou dvě kladná celá čísla. Řekneme, že číslo A je obsaženo v čísle B, jestliže dvojkový (binární) zápis čísla A lze získat vynecháním žádné, jedné nebo několika cifer z dvojkového zápisu čísla B. Pořadí zbývajících cifer přitom zůstane zachováno.

Napište podprogram, který pro zadaná dvě kladná celá čísla A, B určí, zda je A obsaženo v B nebo zda je B obsaženo v A.

- 14) Všechna k-ciferná čísla, v jejichž dekadickém zápisu jsou všechny cifry různé, jsme myšlenkově seřadili podle velikosti. Napište podprogram, který k zadanému číslu této vlastnosti spočte jeho pořadí.

15) Všechna k -ciferná čísla, v jejichž dekadickém zápisu jsou všechny cifry různé, jsme myšlenkově seřadili podle velikosti. Napište podprogram, který k zadanému pořadí určí příslušné číslo.

16) Mějme dánu k -tici ($k \leq 8$) desítkových cifer $c_1, c_2, c_3, \dots, c_k$ (cifry se mohou v k -tici opakovat). Všechna čísla, jejichž desítkový zápis se dá sestavit z těchto cifer (na pořadí nezáleží, není nutno použít všechny) myšlenkově seřadíme do posloupnosti podle velikosti.

Sestavte podprogramy, které

- určí zda číslo má požadovaný tvar
- k číslu N najde N -té číslo dané vlastnosti

17) Mějme dánu k -tici desítkových cifer $c_1, c_2, c_3, \dots, c_k$ $k \leq 8$ (cifry se mohou v k -tici opakovat). Všechna čísla, jejichž desítkový zápis se dá sestavit z těchto cifer (na pořadí nezáleží, není nutno použít všechny) myšlenkově seřadíme do posloupnosti podle velikosti.

Sestavte podprogramy, které

- určí zda číslo má požadovaný tvar
- k číslu dané vlastnosti zjistí, kolikáté je.

18) Napište program, který najde a vypíše všechny latinské čtverce řádu N . Latinský čtverec je matice tvaru $N \times N$ vytvořená z čísel od 1 do N , v jejímž každém řádku i sloupci jsou všechny prvky různé. (N ze vstupu, $N < 10$.)

19) Napište program, který najde všechny diagonální latinské čtverce řádu N (hodnota N vstupuje - $N < 10$). Diagonální latinský čtverec je matice tvaru $N \times N$ vytvořená z čísel od 1 do N , v jejímž každém řádku, sloupci i a na obou úhlopříčkách jsou všechny prvky různé.

20) Na šachovnici o rozměrech $M \times N$ polí jsou některá pole "zakázaná" (např. jsou obsazena). Napište program, který pro zadanou dvojici nezakázaných polí najde nejkratší cestu šachovým králem z prvního pole na druhé. Cesta nesmí vést přes zakázaná pole.

21) Na šachovnici o rozměrech $M \times N$ polí jsou některá pole "zakázaná" (např. jsou obsazena). Napište program, který pro zadanou dvojici nezakázaných polí najde nejkratší cestu šachovou věží z prvního pole na druhé. Cesta nesmí vést přes zakázaná pole.

22) Na šachovnici o rozměrech $M \times N$ polí jsou některá pole "zakázaná" (např. jsou obsazena). Napište program, který pro zadanou dvojici nezakázaných polí najde nejkratší cestu šachovým jezdce z prvního pole na druhé. Jezdec nesmí vstoupit na zakázaná pole.

- 23) Na šachovnici o rozměrech $M \times N$ polí jsou některá pole "zakázaná" (např. jsou obsazena). Je dáno pole, na němž stojí šachový král. Napište program, který najde a vytiskne všechna pole šachovnice, na která může král dojít. Král cestou nesmí vstoupit na žádné zakázané pole.
- 24) Vytiskněte nástěnný kalendář pro zadaný rok. Týdny jsou uspořádány do sloupců a měsíce do čtyř vodorovných řad, každá po třech měsících. Dbejte na úpravnost výstupu.
Stačí se omezit na Gregoriánský kalendář (platí od 15.10.1582). Přestupné roky v Gregoriánském kalendáři jsou takové, které jsou dělitelné čtyřmi, ale ne stem, anebo jsou dělitelné 400.
- 25) Vstupem programu je textový soubor. Slovem nazveme takový úsek textu, který je z obou stran oddělen jednou či více mezerami nebo konci řádků. Napište program, který dostane ze vstupu číslo N a vstupní text vytiskne „srovnaný do bloku“ s délkou řádky N , tj. tak, aby
- Všechny řádky (až na případně poslední) měly délku N a první slovo na nich začínalo v prvním sloupci a poslední končilo v N -tém sloupci.
 - Na každý řádek bylo umístěno co nejvíce slov, které se tam vejdou
 - Počty mezer mezi slovy byly rozděleny rovnoměrně.
- Můžete předpokládat, že ve vstupním souboru nebude žádné slovo delší než 10 znaků a požadovaná délka řádky bude větší než 21.
- 26) Vytvořte podprogramy, které
- zadané celé kladné číslo vypíše slovně (jako na složenku - např. Devětsetdevadesátdevětmiliónů jednotodvatisícedevětsetdevadesátdevět).
 - Ze slovního zápisu čísla spočítá jeho hodnotu.
Stačí se omezit na hodnoty menší než 10^9 .
- 27) Šifra "Nový hrabě Monte Christo". Text se zakóduje do čtvercové matice $N \times N$. Kódovacím klíčem je čtvercová mřížka stejných rozměrů s vhodně vyřezanými otvory. Text se vpisuje do otvorů ve mřížce postupně po řádcích, vždy do každého otvoru jedno písmeno. Pak se mřížka otočí o 90° vpravo a další text se opět vpisuje do otvorů. Toto se opakuje celkem čtyřikrát. Po zaplnění celé matice se mřížka odstraní a obsah matice se vypíše po řádcích. Je-li šifrovaná zpráva delší než jeden čtverec (tj. delší než $N \times N$ písmen), rozdělí se na více úseků, každý o délce 1 čtverce (a případně doplní na potřebný počet lib. znaky). Napište kódovací a dekódovací proceduru.
- 28) Najděte všechny alespoň trojčlenné aritmetické posloupnosti prvočísel menších než vstupující hodnota N ($N \leq 1000$). Vypisujte jen maximální posloupnosti (maximální v tom smyslu, že nejdou prodloužit) a každou právě jednou.

- 29) Textový soubor ADRESAR.ADR obsahuje na každém řádku jedno jméno souboru ve tvaru, jaký používá DOS, tj. vlastní jméno délky 1 až 8 znaků a tečkou oddělená přípona tvořená maximálně třemi znaky. Budeme předpokládat, že jména souborů jsou tvořena pouze písmeny a číslicemi a že soubor ADRESAR.ADR obsahuje nejvýše sto jmen souborů.
- Napište program, který dostane na vstupu specifikaci zvolených souborů, vybere ze seznamu ADRESAR.ADR všechna jména souborů vyhovující zadané specifikaci a vypíše je na obrazovku. Specifikace má stejný tvar, jaký uvádíme např. v příkazu DIR. Má tedy podobu jména souboru, v němž se navíc mohou objevit znaky "?" a "*". Znak "?" zastupuje jeden libovolný znak ve jméně souboru, zatímco znak "*" nahrazuje ve jméně libovolný počet (tedy včetně 0) jakýchkoliv znaků. V jedné specifikaci může být použito i více znaků "?" znak "*" se však může vyskytovat před i za tečkou nejvýše jednou. Přípustné jsou tedy např. specifikace :A*B.E* , ??A*B.E* , ???.*? , ale ne A*b*.EXE
- Pozn.: Nějaký vhodný soubor ADRESAR.ADR si pro potřeby ladění a předvedení programu sami vytvořte pomocí editoru.
- 30) Napište podprogramy, které budou převádět text do morseovky a zpět. Stačí, budete-li umět kódovat písmena a číslice. Je vhodné, aby si program přečetl morzeovku z dat (a mohl tedy pracovat i pro jiný kód tohoto typu) a vytvořil si datové struktury vhodné pro kódování a dekodování – ty nemusí být pro oba směry stejné. Představte si, že vstup bude dlouhý a vyplatí se tedy investovat do rychlosti. Obslužte nějak vhodně znaky, pro něž neznáte kód resp. neznámé kódy.
- 31) Napište podprogramy, které budou umožňovat vstup a výstup celých čísel (hodnot typu integer) římskými číslicemi. Z podprogramů vytvořte jednotku (unit) a napište krátký testovací hlavní program. Stačí se omezit na hodnoty z intervalu 1..5999.
- 32) Napište podprogramy, které budou umožňovat vstup a výstup celých čísel (hodnot typu integer) v poziční soustavě o libovolném základu. Pamatujte na formáty při výstupu. Z podprogramů vytvořte jednotku (unit) a napište krátký testovací hlavní program. (Základ soustavy ≤ 16 , cifry jsou 0123456789ABCDEF.)
- 33) Ve vstupním textovém souboru je uloženo několik (nejméně jedno) dlouhých celých čísel bez znaménka. Každé z čísel má nejvýše 50 cifer a je vždy zapsáno v souboru na samostatném řádku. Počet čísel není předem znám, není jich však více než 50. Napište program, který spočítá součet všech čísel uložených v souboru a výsledek vypíše na obrazovku. Pro potřeby ladění si nějaký vstupní soubor vytvořte v editoru.
- 34) Spočítejte přesně součet, rozdíl, součin a celočíselný podíl dvou dlouhých celých čísel. Čísla jsou zadána na vstupu, mohou (ale nemusí) obsahovat znaménko, každé z nich má nejvýše 300 cifer.

- 35) Napište program, který spočte přesně faktoriál daného velkého čísla (např. 200!). Zvolte vhodné omezení na maximální přípustnou hodnotu daného čísla (např. 200).
- 36) Program načte d,m,r - datum (den, měsíc, rok) a určí, o jaký den v týdnu se jedná. (Nepřestupný rok má 365 dní, přestupný 366; přestupný je takový rok, jehož číslo je dělitelné 4 a není dělitelné 100 nebo je dělitelné 400).
- 37) Program načte d,m,r - datum (den, měsíc, rok), číslo N a určí, jaké datum bude za N dní od zadaného data. (Nepřestupný rok má 365 dní, přestupný 366; přestupný je takový rok, jehož číslo je dělitelné 4 a není dělitelné 100 nebo je dělitelné 400)
- 38) Maharádža je figurka z pohádkového šachu, která si v každém tahu může vybrat, zda bude táhnout jako dáma nebo jako jezdec.
Pneumatika $M \times N$ je hrací plocha, která vznikne z šachovnice $M \times N$ tím, že slepíme její pravý kraj s levým a horní s dolním.
 Najděte pro zadané rozměry pneumatiky M a N nejmenší počet maharádžů K , který je schopen ohrozit všechna pole pneumatiky. Program by měl (volitelně) být schopen ukazovat postupně všechna různá umístění K maharádžů, při kterých ovládají celou pneumatiku.
- 39) Na vstupu je textový soubor rozdělený mezerami a konci řádek na slova.
 Vytvořte výstupní textový soubor, který bude formátován (slova, řádky) přesně stejně jako soubor vstupní a při tom v něm budou provedeny následující „cenzurní zásahy“ :
 Každé slovo, pro nějž alespoň jedno z tří po něm bezprostředně následujících slov končí na „ova“, bude nahrazeno hvězdičkami.
 Vstupní soubor smíte číst jen jednou.
- 40) Na vstupu je textový soubor rozdělený mezerami a konci řádek na slova. Vyskytují se v něm jen písmena anglické abecedy.
 Vytvořte výstupní textový soubor, který bude formátován (slova, řádky) přesně stejně jako soubor vstupní a při tom v něm budou provedeny následující „cenzurní zásahy“ :
 Každé slovo, jehož některé ze (dvou) sousedních slov obsahuje x -tý výskyt některé samohlásky v souboru, kde x je prvočíslo, bude nahrazeno hvězdičkami. Výskyty každé samohlásky se počítají zvlášť a nehledí se při tom malá a velká písmena – výskytu znaků [u,U] jsou tedy výskyty samohlásky u, výskyty znaků [e,E] jsou výskyty samohlásky e.
 Vstupní soubor smíte číst jen jednou, můžete předpokládat, že obsahuje méně než 25000 slov.
- 41) Napište program pro počítání velkých mocnin matic (např. 1992. mocnina matice). Prvky matic jsou čísla typu real.

- 42) Vytvořte program, který vytiskne frekvenční tabulku obsahující N nejčastěji se vyskytujících slov v zadaném vstupním souboru uspořádanou podle počtu jejich výskytů. Slovem rozumíme libovolnou posloupnost písmen anglické abecedy, slova jsou oddělena mezerami resp. konci řádků. Pro účely porovnávání slov zanedbáváme rozdíl mezi malými a velkými písmeny (považujeme tedy slova Adela, adela, aDELa a ADELa za stejná). Můžete předpokládat, že $N < 100$ a že ve vstupním souboru je nejvýše 1000 různých slov. Příklad možného výstupu pro $N=5$:

a	123
ale	54
br	54
klasicky	27
co	13

- 43) Řekneme, že úsek číselné posloupnosti je 1-rostoucí, jestliže z něj jde vynecháním nejvýše jednoho čísla vytvořit rostoucí posloupnost. Na vstupu je textový soubor obsahující čísla typu integer (můžete předpokládat, že na každé řádce je právě jedno číslo). Nalezněte nejdelší 1-rostoucí úsek vstupující posloupnosti (tj. pořadové číslo jeho prvního členu, jeho délku a údaj zda je třeba nějaký prvek vypustit – a pořadové číslo tohoto prvku). Vstupní soubor smíte číst jen jednou.