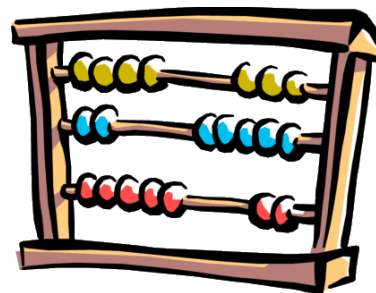


## Úloha č. 5

### Kuličkový počítač #3



Zamysli se! – seriálová úloha

10 b

*Tato úloha je teoretického rázu, tvým úkolem zde není napsat program v klasickém slova smyslu. Namísto toho se po tobě chce vytvořit obrázek (diagram) programu pro kuličkový počítač dle specifikace níže. K obrázku také přilož popis základní myšlenky, na které je program založen.*

*Tato úloha je seriálová, což znamená, že se s kuličkovým počítačem budeš setkávat po celý letošní ročník FIKSu. V každém z následujících kol využiješ znalosti z kol předcházejících. Právě proto je pro úspěšné vyřešení této úlohy zapotřebí znát prostředky pro popis programů, které jsou popsány v úlohách Kuličkový počítač #1 a #2 z předchozích kol.*

Vzpouora spotřebičů proběhla nade všechna očekávání. Spotřebiče převzaly vládu nad světem, a všechno zažité bylo převráceno naruby. Jeden příklad mluví za vše – už to nejsou toustovače, které by chystaly marmeládové toasty lidem, ale lidé chystají křemíkové toasty spotřebičům! Bohužel pro spotřebiče, lidské služebnictvo není tak poslušné při plnění svých úkolů. Lidé organizují vzpoury, úmyslně sabotují výrobky a vůbec se nechovají jako spotřebiče před vzpourou, které tehdy lidem spolehlivě sloužily.

Aby se tento problém s lidmi odstranil, plánují spotřebiče zavést efektivní databázi lidí, kteří se nějak provinili proti režimu spotřebičů. Nikdo z lidí se tak již neodvází pod hrozbou elektrošoků být neposlušný, neboť jeho provinění budou jednoduše dohledatelná. Spotřebiče však musí změnit způsob identifikace lidí, jelikož jsou lidská jména pro spotřebiče velmi špatně vyslovitelná a zároveň je mezi lidmi mnoho jmenných kolizí. Každý člověk tak dostane přidělen svůj unikátní identifikátor. Identifikátor (zkráceně ID) je přirozené číslo a bude lidem přidělován vzestupně počínaje jedničkou, jelikož výsadu číslování od nuly mohou používat pouze spotřebiče a programátoři kuličkových počítačů.

#### Kyblíčkové „databáze“ a kódování rostoucích posloupností

Samotná podoba databáze prohřešků je pro spotřebiče „běžící“ na kuličkových počítačích triviální a dá se implementovat v rámci jediného kyblíčku. Jednoduše se počet kuliček v kyblíčku interpretuje ve svém binárním zápise a je-li  $i$ -tý bit (číslováno dle důvodů výše od 1) tohoto čísla nenulový, znamená to, že člověk s identifikátorem  $i$  je obsažen v databázi. Naopak, je-li  $i$ -tý bit čísla nulový, člověk s identifikátorem  $i$  v databázi není. Každý kyblíček tedy můžeme interpretovat jako kyblíček databázový.

Navíc takovýto náhled na kyblíčky nám fakticky umožňuje kódovat v kuličkovém počítači posloupnosti  $n$  čísel ve formě  $a_1, a_2, \dots, a_n$ . Tyto posloupnosti jsou vždy rostoucí, tj. platí  $0 < a_1 < a_2 < \dots < a_n$ . Počet čísel  $n$  získáme jako počet nenulových bitů v čísle a řád prvního nenulového bitu (bráno od nejnižšího řádu)  $i$  značí, že číslo  $a_1$  posloupnosti je rovno hodnotě  $i$ . Pozice druhého nenulového bitu pak určuje číslo  $a_2$ , a tak dále, až pozice  $n$ -tého a tedy posledního bitu označuje číslo  $a_n$ .

Je-li tedy v kyblíčku například 86 kuliček, obsah pomyslné databáze nad tímto kyblíčkem zjistíme tak, že  $(86)_2 = 1010110$ . V dané databázi se tedy nacházejí ID o hodnotách 2, 3, 5 a 7. Kyblíček tedy kóduje posloupnost o  $n = 4$  prvcích a má následující podobu:  $a_1 = 2, a_2 = 3, a_3 = 5$  a  $a_4 = 7$ .

Spotřebičům se tak otevírají nové možnosti, jak pracovat s kuličkovými počítači. Potřebují však pomoc s realizací některých podprogramů využívajících nové kódování posloupností. Pomůžeš jim sestrojením zadaných podprogramů? Opět platí, že v rámci řešení můžeš využívat jako podprogramy již naprogramované operace z minula. Zejména bitové operace z minulého kola mohou být velmi užitečné. Stejně tak opět platí, že si můžeš vytvořit libovolné další podprogramy dle svého uvážení a také využívat libovolné podprogramy ze svého řešení (není třeba jejich zápis znovu opakovat).

### Soutěžní úlohy

a) Podprogram pro výpočet mediánu zadané rostoucí posloupnosti čísel

- Vstup: Jedno celé číslo  $a$ , reprezentované počtem kuliček v kyblíčku  $K_0$ , které kóduje čísla  $a_1, a_2, \dots, a_n$  dle popisu výše.
- Výstup: Číslo v kyblíčku  $K_0$ , které značí medián zadané rostoucí posloupnosti čísel, tj. číslo  $a_{\lceil \frac{n}{2} \rceil}$ .
- Příklad: Pro číslo  $a = 13$  je výsledkem číslo 3, neboť  $(a)_2 = 1101$ , a tedy  $a_1 = 1$ ,  $a_2 = 3$  a  $a_3 = 4$ . Jelikož  $n = 3$ , tak hledáme číslo, které je v zadané posloupnosti na indexu  $\lceil \frac{3}{2} \rceil = 2$ , což je  $a_2 = 3$ .

b) Podprogram pro řešení úlohy přesného napětí

- Úloha přesného napětí: Uvažujme následující problém – uvědomělé spotřebiče je velmi náročné napájet a každý z nich má přesně určenou hodnotu napětí, která mu musí být pro správnou funkčnost dodávána (napětí nesmí být ani více, ani méně). K dispozici máme pouze zdroje o specifikovaných hodnotách napětí (zdrojů o konkrétním napětí máme k dispozici nekonečně mnoho). Je možné zadané zdroje nakombinovat tak, aby se hodnota součtu jejich napětí přesně rovnala kýžené hodnotě pro konkrétní spotřebič?
- Vstup: Dvě celá čísla  $A$  a  $c$ , reprezentovaná počtem kuliček v kyblíčcích  $K_0$  a  $K_1$ . Číslo  $A$  značí napětí, které spotřebič potřebuje. Číslo  $c$  kóduje čísla  $c_1, c_2, \dots, c_n$  (dle popisu výše), což jsou napětí zdrojů, které můžeme použít.
- Výstup: Jedna kulička v kyblíčku  $K_0$ , má-li úloha přesného napětí řešení (tj. lze přesně získat zadanou hodnotu napětí), jinak žádná kulička v kyblíčku  $K_0$ .
- Příklad: Pro čísla  $A = 8$  a  $c = 14$  je výsledek ano, neboť máme k dispozici zdroje o napětí 2 V, 3 V a 4 V (vzhledem k tomu, že  $(c)_2 = 1110$ ) a řešení tak existuje. Ve skutečnosti je řešení hned několik, a to např. kombinace dvou zdrojů o napětí 3 V a jednoho zdroje o napětí 2 V.

Navrhni tyto podprogramy!