

Univerzita Karlova
Přírodovědecká fakulta



Doprovodný dokument

k programu

Převod čísla c , $c \in \mathbb{Z}^+$, z dvojkové soustavy do šestnáctkové a naopak

Úvod do programování
MZ370P19

Filip Zadražil

3. BGEKA

Chýnov, 4. 2. 2021

1) Zadání

Hlavním úkolem tohoto programu je uživatelem vybrané kladné celé číslo ve dvojkové soustavě převést do soustavy šestnáctkové a následně uživatelem vybrané číslo v šestnáctkové soustavě převést do soustavy dvojkové. Zároveň má být program ošetřen proti nevalidním vstupům, kterými jsou jakékoli hodnoty, které nejsou v zadané množině čísel dvojkové (resp. šestnáctkové) soustavy.

2) Zvolený algoritmus

Při volení algoritmů bylo nejefektivnější využít vestavěných funkcí Pythonu. Nebylo tedy třeba importovat žádné knihovny třetích stran. Byly využity následující dva algoritmy:

- **bin()**
je vestavěná funkce Pythonu, která převádí čísla typu Integer do binárního textu prefixovaného „0b“. Tento prefix je možné odstranit různými logikami.
- **hex()**
je též vestavěná funkce Pythonu, která převádí čísla typu Integer do lowercase textu prefixovaného „0x“. Tento prefix je možné odstranit různými logikami.

3) Potenciální algoritmy

Pokud bychom uvažovali nad prostou metodou konverze hexadecimální do binární soustavy, dalo by se použít například jednoduchou logiku srovnávání, rozložit hexadecimální input do čísel a písmen a následně definovat například $a == 1010$, $b == 1011$, ... a následně jednotlivě přidělit tyto proměnné existujícím písmenům a číslům v inputu.

4) Program

Struktura programu je poměrně prostá. Je využíváno cyklu *while*, který na úplném začátku vyzve uživatele k zapsání vstupní hodnoty. Po zapsání se cyklus snaží o převod hodnot do dvojkové/šestnáctkové soustavy pomocí funkce *bin()/hex()*. Naráží však na podmínky. Tou první je podmínka kladnosti čísla. Druhá podmínka zajišťuje zamítnutí převodu každého čísla, které nemá základ 2 respektive 16, čili nepatří do dvojkové resp. šestnáctkové soustavy. Obě tyto podmínky končí příkazem *continue*, který zajistí, že po zadání nevalidní hodnoty program neskončí, ale vyzve uživatele pro vložení nové, validní hodnoty. Třetí podmínkou je samotné užití výrazu *int()* při převodu, která zabraňuje vstupu necelých čísel.

Nutno zdůraznit, že funkce *hex()* a *bin()* outputují text s již zmíněným prefixem. Ten vždy zabíral první dvě pozice v outputu. Tudíž bylo nejjednodušší odebrat první dva znaky těchto mezivýsledků. V oficiální dokumentaci se jako řešení zmiňuje funkce *format(value, [format_spec])*.

5) Reprezentace vstupních a výstupních dat

Vstupem do tohoto programu je hodnota, kterou si uživatel sám zvolí. Program k tomu uživatele vybízí větou: *"Napište libovolné celé kladné číslo v binární/hexadecimální soustavě:"*. V případě dvojkové soustavy to musí být celé číslo obsahující pouze jedničky a nuly. Pro šestnáctkovou soustavu platí, že zvolené celé číslo může obsahovat pouze různé kombinace čísel 1-9 a zároveň písmen a-f.

Výstupem je pak hodnota, která je ekvivalentem čísla vloženého, ale je převedená do opačné soustavy. Výsledná hodnota je pak vytisknuta v rámci věty: *„Ekvivalentní hodnota*

v *hexadecimální/binární soustavě je *výsledná hodnota**." Tento celkový výstup programu je zobrazen v terminálu a není nikam následně ukládán.

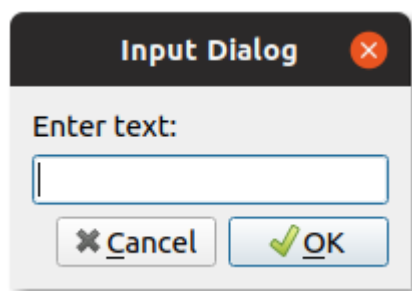
6) Průběh práce

Práce na tomto programu probíhala převážně hladce. O užití funkcí `bin()` a `hex()` (zajišťujících samotný převod čísel) bylo rozhodnuto hned v samém začátku sestavování programu, neboť se jejich použití jeví jako velice rychlé, přehledné a efektivní. Nejprve došlo k vytvoření právě samotných převodů čísel a až dodatečně byly do programu přidány části, které kontrolují validitu jednotlivých vstupů. Právě zakomponování těchto prvků byl největší kámen úrazu vytvoření programu. Jedním ze zádrhelů byl i nulový vstup. Ten byl způsobený srovnáváním textového vstupu „0“ s integerem 0 bez uvozovek. Po přidání uvozovek a úpravě dalších drobných nesrovnalostí se však nakonec podařilo dosáhnout plné funkčnosti programu.

7) Možná vylepšení programu

Myslím, že možností, jak program technologicky vylepšit/zdokonalit již není mnoho. Jako příklad bych však uvedl, že jistým přínosem pro program by bylo vytvoření nové vlastní podmínky pro vstupní hodnotu, která je zadána neceločíselná. V současnosti totiž program na tento chybný vstup reaguje hláškou: *„Vámi zadané číslo není v binární/hexadecimální soustavě nebo není celočíselné. Zkuste to, prosím, znovu!“*, což není zcela přesné vyjádření a dalo by se jinak definovat.

Nepochybně dobré vylepšení by bylo žádat o input z nějakého uživatelsky přívětivého okénka (dialog boxu) pomocí knihoven třetích stran, jako například:



8) Závěr

Program bylo velice zajímavé utvářet a jsem rád za jeho výběr. Při převodu z hexadecimální soustavy mě napadlo využití na úrovni výpočetní techniky. Jelikož počítače využívají ke svým operacím pouze binární soustavu, je třeba převod například barev ve formátu `#ffffff` do binární soustavy. Tato operace je ve své podstatě triviální ale velmi důležitá a užitečná. Byť mě nenapadá žádné další konkrétní praktické využití, jistě by mnozí jeho „služeb“ rádi využili. I na internetu je totiž k dohledání několik online převodníků mezi těmito soustavami, které pracují na podobném principu. To napovídá, že k podobným „mezisoustavovým“ převodům dochází častěji, než by se mohlo zdát.