B-I-4 Reálne čísla

V tejto súťažnej úlohe sa zoznámime s tým, ako si počítače poradia s reálnymi číslami. Presnejšie, budeme sa zaoberať najbežnejším formátom používaným na uloženie reálnych čísel v počítači: formát *s plávajúcou bodkou* (floating-point representation). Tieto divne znejúce slová znamenajú to, že si počítač samostatne pamätá cifry čísla a samostatne miesto, kde je hranica medzi celou a necelou časťou.[1] Celé to teda bude fungovať podobne, ako keď číslo 1 234 567 zapíšeme v tvare 1.234567×10^6.
Samozrejme, hneď prvý rozdiel je v tom, že počítače nepoužívajú desiatkovú sústavu ale dvojkovú – teda jediné dve cifry sú 0 a 1. V nasledujúcom texte preto budeme občas aj my používať dvojkovú sústavu. Na miestach, kde nebude jasné, ktoré čísla sú v dvojkovej a ktoré v desiatkovej sústave, použijeme dolné indexy. Napr. 11_{10} je jedenásť, zatiaľ čo 11_2 je tri.

To, že počítače používajú dvojkovú sústavu namiesto desiatkovej ich nijak neobmedzuje. Totiž podobne ako v desiatkovej sústave, aj v dvojkovej vieme zapisovať aj necelé čísla: 0.1_2 je jedna polovica, 0.01_2 je jedna štvrtina, a tak ďalej. Teda napríklad $0.1001_2 = 0.5625_{10}$.
Samozrejme, niektoré reálne čísla majú konečný zápis, zatiaľ čo iné nie. Všetky iracionálne čísla, ako napríklad π, e, alebo $\sqrt{2}$ majú aj v dvojkovej sústave nekonečný zápis, ktorý navyše nie je periodický. Ako sme už videli vyššie, niektoré racionálne čísla majú v dvojkovej sústave konečný zápis. Ale nie všetky. Napríklad jedna tretina v dvojkovej sústave je $0.\overline{01} = 0.01010101\ldots$
Zaujímavé je tiež, že každé racionálne číslo, ktoré má konečný zápis v dvojkovej sústave, má konečný zápis aj v desiatkovej sústave. (Viete, prečo je to tak? Skúste si to dokázať.) Naopak to však neplatí. Napríklad jedna pätina je v desiatkovej sústave 0.2, zatiaľ čo v dvojkovej je to $0.\overline{0011}$.

Dvojková sústava je tiež zaujímavá tým, že zápis úplne každého čísla okrem nuly začína jednotkou. Každé číslo okrem nuly teda vieme v dvojkovej sústave zapísať v nasledovnom tvare: $(\pm 1.m \times 2^e)$. V tomto zápise m (nazývané *mantisa*) predstavuje ľubovoľnú, konečnú alebo aj nekonečnú, postupnosť cifier, zatiaľ čo e (nazývané *exponent*) je nejaké celé, možno aj záporné číslo. Napríklad pre číslo $x = 5.125_{10} = 101.001_2$ tento zápis vyzerá nasledovne: $x = 1.01001 \times 2^2$. Pre číslo $y = -0.125_{10} = -0.001_2$ máme $y = -1. \times 2^{-3}$.
Keď sa reálne čísla ukladajú do pamäte počítača vo formáte s plávajúcou bodkou, používame presne tento formát, s jedinou zmenou: pamäť počítača je bohužiaľ konečná. Preto v tomto formáte nevieme reprezentovať ani zďaleka všetky reálne čísla. Presne vieme uložiť len tie, ktoré majú v dvojkovej sústave konečný a dostatočne krátky zápis (t.j. krátku mantisu) a zároveň dostatočne malý exponent. Ak chceme uložiť nejaké iné číslo, máme smolu. Najlepšie, čo vieme spraviť (a čo sa aj v skutočnosti v počítači stane) je, že nastane *zaokrúhľovacia chyba* a namiesto skutočnej hodnoty si počítač zapamätá najbližšiu hodnotu, ktorú skutočne uložiť vie. Inými slovami, zapamätá si z mantisy len toľko *najvýznamnejších bitov*, koľko sa mu zmestí do pamäte.[2]
Ak si teda v takomto formáte uložíme hodnotu $4/3 = 1.\overline{01}$, nebude v pamäti počítača uložená presná hodnota štyroch tretín, ale nejaké číslo, ktoré sa od nich o maličkú trošku líší. Ak by sme napríklad v pamäti mali miesto len na sedem bitov mantisy, uložili by sme si namiesto presných štyroch tretín číslo 1.0101011×2^0. Všimnite si, že uložená mantisa (0101001) končí cifrou 1, keďže toto číslo je ku štyrom tretinám bližšie ako 1.0101010×2^0.

Dva najbežnejšie používané verzie formátu s plávajúcou bodkou sa volajú *single precision* a *double precision* (stručne len *single* a *double*, voľne preložené: obyčajná a dvojnásobná presnosť). Číslo uložené vo formáte single zaberá 4 bajty pamäte, a to nasledovne: 1 bit je znamienko, potom 8 bitov exponent, a na záver 23 bitov tvorí mantisa. Číslo vo formáte double zaberá dvakrát toľko pamäte, teda 8 bajtov. Opäť je tam jeden bit na znamienko, potom máme 11-bitový exponent a na záver 52-bitovú mantisu.
Bežné formáty „reálnych“ čísel v programovacích jazykoch zvyknú zodpovedať týmto dvom verziám: single precision je napríklad `float` v Jave aj C++ a `single` vo FreePascale, double precision je napríklad `double` v Jave, C++, aj FreePascale a `float` v Pythone.

[1] V slovenčine v desiatkovej sústave toto miesto označujeme desatinnou čiarkou. V anglicky hovoriacom svete a preto aj v programovacích jazykoch sa však používa bodka – to je ten „point“ z názvu.

[2] Drobný technický detail: Treba sa dohodnúť, čo sa má stať, ak sa presná hodnota, ktorú si chceme uložiť, nachádza presne v strede medzi dvoma hodnotami, ktoré si vieme v pamäti počítača reprezentovať. My budeme v tomto zadaní v súlade so štandardom predpokladať, že takúto hodnotu vždy zaokrúhlime na „okrúhlejšie“ číslo – inými slovami, že v takomto prípade sa vyberie tá možnosť, pri ktorej uložená mantisa končí nulou.

Všimnite si, že pri ukladaní čísla v tvare $(\pm 1.m \times 2^e)$ ukladáme ako mantisu len m. Jednotku pred ňou ukladať netreba, keďže tú majú všetky čísla rovnakú. Ak teda máme 23-bitovú mantisu, znamená to, že si do pamäte uložíme *24 najvýznamnejších bitov* ukladaného čísla. A keďže $2^{24} > 10^7$, neformálne to znamená, že v desiatkovej sústave by sa nám aj napriek zaokrúhľovacím chybám malo zachovať aspoň sedem najvýznamnejších cifier. Ak by sme napríklad do premennej typu single uložili hodnotu 1.234567890_{10}, po zaokrúhlení by sme dostali presnú hodnotu 1.2345678806304931640625.
Pre formát double je 2^{53} o čosi menej ako 10^{16}. Ak teda reálne číslo reprezentujeme vo formáte double, chyba sa pohybuje na úrovni zhruba sedemnástej cifry v jeho desiatkovom zápise.

Odovzdávanie riešení

Toto je teoretická úloha. Spísané riešenie vo formáte PDF odovzdajte pomocou webového rozhrania.

TRIDSIATY TRETÍ ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Michal Forišek
Recenzia: Michal Forišek
Slovenská komisia Olympiády v informatike
Vydal: IUVENTA – Slovenský inštitút mládeže, Bratislava 2017