Těžký příklad — Sčítání a odčítání reálných čísel v zadané číselné soustavě

1 Zadání

- Napište program **base.py**, který sečte dvě zadaná čísla a od výsledku odečte třetí číslo v zadané číselné soustavě.
- **Vstup:** čtyři řádky standardního vstupu: první řádka definuje základ číselné soustavy, druhá, třetí a čtvrtá obsahují tři desetinná čísla v zadané číselné soustavě
  - první řádka definuje základ číselné soustavy
  - druhá, třetí a čtvrtá obsahují tři desetinná čísla v zadané číselné soustavě
- **Výstup:** číslo v zadané soustavě, které je součtem prvních dvou čísel od kterých je odečteno číslo třetí nebo "ERROR" pokud jsou čísla zadána chybně
  - Řešení vypisujte bez tzv. **leading/trailing zeros**
    - příklad leading zeros: "00.1" (správně má být "0.1"), "-010.1" (má být "-10.1")
    - příklad trailing zeros: "0.00100" (správně je "0.001"), "-0.620" (správně je "-0.62")

Program v souboru **base.py** odevzdejte pomocí odevzdávacího systému (úloha HW02).

2 Poznámky

- Základ výstupní soustavy je v rozmezí 2 .. 36, správnost základu soustavy (první vstup) nemusíte testovat.
- Pokud je na vstupu špatně zadané číslo (obsahuje jiné znaky než takové, které jsou přípustné pro zadanou soustavu a znak `.`, případně obsahuje znak `.` vícekrát), pak program vytiskne na výstup "ERROR"
  - Pro soustavy o základu menším nebo rovno 10, obsahuje desetinné číslo pouze čísla od 0 .. `základ soustavy - 1` a znak `.`
  - Pro soustavy o základu větším než 10, obsahuje desetinné číslo na výstupu čísla 0,..,9 a malá písmena a, .. , z taková, že hodnota písmena `znak` splňuje $ord(znak) - ord('a') < zaklad\_soustavy - 10$.
- Např. desetinné binární číslo $101.0101 = 1*2^2+0*2^1+1*2^0+0*2^{-1}+1*2^{-2}+0*2^{-3}+1*2^{-4} = 5.3125$.
- Výsledek počítejte v zadané soustavě, **nepřevádějte** čísla do desítkové soustavy a zpět.
- Pokud používáte odčítání ze základní školy, tak při odčítání musíte vždy odčítat menší číslo od většího.

3 Příklady

3.1 Příklad 1.

Vstup programu je:

```
3
1.2222
1.0121
2.12
```

Výstup programu bude:

```
0.122
```

Čísla jsou zapsána správně pod sebou, takže můžete použít písemné sčítání a po sečtení prvních dvou čísel získáte 10.012. Po odečtení třtího čísla získáte výsledek 0.122.

3.2 Příklad 2.

Vstup programu je:

```
33
pm.ttnp1
l.e12w
n.m6hqnq
```

Výstup programu bude:

```
pk.lo8ua7
```

Tento příklad je již složitější, ale Váš program by měl dát stejný výledek.

3.3 Příklad 3.

Vstup programu je:

```
2
1.10011
10.011
1101.0011
```

Výstup programu bude:

```
-1001.00111
```

V tomto případě je po součtu dvou prvních čísel výsledek `11.11111`, což je číslo menší než třetí číslo `1101.0011`. Z tohoto důvodu bude výsledek záporný. Absolutní hodnotu výsledku získáte jako rozdíl většího čísla a menšího, tedy `1101.0011 - 11.11111`. Výsledek pak vytiskněte s počátečním znaménkem mínus `-1001.00111`.

3.4 Příklad 4.

Vstup programu je:

```
4
10.1313
11.2302214
23021.331
```

Výstup programu bude:

```
ERROR
```

V tomto případě je chybně zadáno druhé reálné číslo `11.2302214` které obsahuje cyfru 4, která nemá smysl v soustavě o základě 4.