

Lehký příklad — "Rozdíl časů" (6 bodů)

Zadání:

- Napište program `time_diff.py`
- Na standardním vstupu máte zadány dva časy, na každé řádce jeden, ve formě např. 15h24m30s. Správný formát vstupu a výstupu je [dvě cifry <24]h[dvě cifry <60]m[dvě cifry <60]s.
- Standardní vstup obsahuje vždy dvě řádky, ale čas může být zadán chybně.
- Váš program vypočte absolutní hodnotu rozdílu těchto časů a vytiskne tuto hodnotu ve stejném formátu jako vstup na standardní výstup.
- V případě chybného zadání vytiskne ERROR.

Bodování:

- Správné vstupy — max 4b. Testujeme funkčnost pro několik různých kombinací dvojic správně zadaných časů, přidělené body odpovídají procentu správných výsledků.
- Chybné vstupy — max 2b. Opět testujeme několik různých kombinací dvojic vstupních řádků z nichž alespoň jeden je chybný. Přidělené body odpovídají procentu správných výsledků (tj. ERROR).

Příklady:

Vstup:

```
12h45m10s
01h00m00s
```

Výstup:

```
11h45m10s
```

Vstup:

```
10h15m59s
11h15m59s
```

Výstup:

```
01h00m00s
```

Vstup:

```
1h45m10s
01h00m00s
```

Výstup

```
ERROR
```

Vstup:

```
pulnoc
01h00m00s
```

Výstup

```
ERROR
```

Těžký příklad — “Přelévání nádob” (14 bodů)

Zadání:

- Napište program `bin.py`
- Na první řádce standardního vstupu jsou zadány tři celá kladná čísla, které reprezentují objemy tří nádob.
- Na druhé řádce vstupu je zadáno jedno celé kladné číslo, které chceme získat jako konečný objem v libovolné z nádob.
- Program načte tyto dvě řádky vstupu a vytiskne na standardní výstup postup, jak získat zadané množství tekutiny z počátečního stavu, kdy jsou všechny nádoby prázdné.
- Postup se bude skládat z několika řádek ve tvaru:

`Nx` - napustí celou nádobu `x`, hodnota `x` je v rozsahu (0,1,2)

`Vx` - vylij celou nádobu `x`, hodnota `x` je v rozsahu (0,1,2)

`xPy` - objem nádoby `x` přelij do nádoby `y`, hodnota `x` a `y` je v rozsahu (0,1,2). Pokud je objem nádoby `y` menší než množství vody v nádobě `x`, pak po přelití bude nádoba `y` plná a v nádobě `x` zbyde rozdíl objemu vody a objemu nádoby `y`. V opačném případě se veškerý objem nádoby `x` přelije do nádoby `y`.

- Všechny vstupy pro váš program jsou zadány korektně, nemusíte kontrolovat jejich správnost.
- Všechny testovací případy lze vyřešit do 10 kroků.
- Časový limit na vyřešení všech testovacích úloh (cca 15 úloh) je 30 s!

Bodování:

- Program umí řešit i velmi triviální případy (např. jedna z nádob má shodný objem s požadovaným) — max 2b.
- Řešení pro lehké úlohy, kdy stačí provést dvě operace — max 4 b.
- Řešení všech ostatních případů — max 8 b.

Příklad I:

Vstup:

```
2 5 9
6
```

Výstup:

```
N0
0P1
N2
2P1
```

Vysvětlení:

- počáteční stav systému je $[0,0,0]$ - všechny nádoby jsou prázdné
- po provedení příkazu `N0` se stav změní na $[2,0,0]$ - první nádoba je plná
- po provedení příkazu `0P1` se stav změní na $[0,2,0]$ - celou nádobu 0 přeliji do nádoby 1
- po provedení příkazu `N2` se stav změní na $[0,2,9]$
- po provedení příkazu `2P1` se stav změní na $[0,5,6]$
- poslední nádoba nyní obsahuje objem o velikosti 6

Pokračování na druhé straně!

Příklad II:

Vstup:

```
2 5 17
16
```

Výstup:

```
N0
N2
OP1
N0
OP1
2P1
```

Nápověda:

- Zkuste testovat, jestli jedna z nádob nemá stejný objem jako je požadován na vstupu.
- Napište si funkci, která ze současného stavu vygeneruje všechny možné stavy s nalitím, vylitím a přelitím
- Se stavem si pamatujte i jak jste se k tomuto stavu dostali, třeba i rovnou jako výstupní řetězec