

Lehký příklad — analýza odečtů elektroměru

Energetická společnost vám poskytla data o spotřebě elektriny za jeden den. Data jsou ve formě seznamu hodnot, kde každá hodnota odpovídá průměrné spotřebě za určitý časový úsek (X minut), kde X je konstantní pro celý den. Každá hodnota představuje průměrnou činnou výkonovou hladinu v kW za daný X-minutový úsek. Cílem je vytvořit jednoduchý analytický nástroj, který z těchto dat vytáhne praktické informace, které mohou být užitečné pro zákazníky i pro energetickou společnost.

- Napište program **analyze.py**, který:
- **Vstup:** jeden řádek na standardním vstupu o proměnném počtu hodnot oddělených čárkou.
 - Každá hodnota je desetinné číslo v kW (např. 0.43) a představuje průměrnou spotřebu v kW za Xminutový úsek.
 - Počet hodnot je takový, aby pokryl celý den (24 hodin).
 - Protože jsou časové odečty prováděny různými technikami, může se vstupní délka dat pohybovat od 4 hodnot až po 1440.
- **Výstup:** na standardní výstup vypište:
 - řádek 1 — největší skok mezi sousedními měřeními včetně času který k němu náleží.
 - * Výstupem je hodnota skoku s přesností na 3 desetinné místa doplněna o časový interval ve formátu HH:MM-HH:MM, kde první čas je začátek intervalu a druhý čas je konec intervalu.
 - * Pokud je více stejných skoků, vypište ten první.
 - * Příklad výstupu je následující: **1.631 between 18:00-19:00** (Použijte funkci `round()`)
 - řádek 2-4 — tři nejvyšší hodnoty spotřeby a čas kdy nastaly.
 - * Pokud je více hodnot stejných, seřaďte je podle času, kdy nastaly (dřívější hodnoty dříve).
 - * Příklad výstupu je následující: **2.546 at 10:01**
 - řádek 5 — nejdelší „stabilní“ interval – souvislý úsek, kde se hodnoty liší od průměru o nejvýše 10 procent.
 - * Pozor: I všechny podintervaly musí splňovat tuto podmínku!
 - * Výstupem je časový interval ve formátu HH:MM-HH:MM, kde první čas je začátek intervalu a druhý čas je konec intervalu.
 - * Jako konec intervalu se bere čas na konci posledního měření, které do intervalu spadá, plus X minut, které měření pokrývá.
 - * Pokud je více intervalů stejné délky, vypište ten první.
 - * Příklad výstupu je následující: **00:00-02:15**
 - Pokud vstupní data obsahují záporné hodnoty, nebo nan, vypište na standardní výstup pouze řetězec "ERROR" (bez uvozovek) a ukončete program.
- **Poznámky:**
 - Délku časového úseku X zjistíte podle počtu hodnot N na vstupu.
 - Samotný časový úsek je zakódován tak, že začíná v 00:00 a končí v 24:00.
 - * Pokud je X=60 (tj. na vstupu je 24 čísel), pak je první hodnota za 00:00-01:00, druhá za 01:00-02:00, atd.
 - * Pokud je X=30 (tj. na vstupu je 48 čísel), pak je první hodnota za 00:00-00:30, druhá za 00:30-01:00, atd.
 - * Pokud je X=15 (tj. na vstupu je 96 čísel), pak je první hodnota za 00:00-00:15, druhá za 00:15-00:30, atd.
 - Pro výpis času doporučujeme použít `"{:02d}:{:02d}".format(hours,minutes)`, tj. např. `print("{:02d}:{:02d}".format(12,7))` poskytne výpis 12:07

1 Příklady

1.1 Příklad

Vstup

```
1.67,1.338,3.146,0.495
```

Výstup programu

```
2.651 between 12:00-18:00
3.146 at 12:00
1.67 at 00:00
1.338 at 06:00
00:00-06:00
```

Komentář: Protože máme pouze 4 vstupní hodnoty (1.67, 1.338, 3.146, 0.495), znamená to, že každá hodnota pokrývá 6 hodin (24 hodin / 4 hodnoty = 6 hodin na hodnotu). Největší skok je mezi 18:00 až 24:00 (2.651 kW). Tři nejvyšší hodnoty jsou 3.146 kW v 12:00, 1.67 kW v 00:00 a 1.338 kW v 06:00. Nejdelší stabilní interval je od 00:00 do 06:00 (6 hodin).

1.2 Příklad

Vstup

```
0.4,0.4,0.43,0.34,0.41,0.39,0.38,0.56,1.42,0.98,1.27,1.39,1.19,2.37,1.01,0.98,0.82,0.61,
0.43,0.43,0.4,0.43,0.39,0.42
```

Výstup programu

```
1.36 between 13:00-14:00
2.37 at 13:00
1.42 at 08:00
1.39 at 11:00
18:00-24:00
```

Komentář: Zde máme 24 vstupních hodnot, což znamená, že každá hodnota pokrývá 1 hodinu (24 hodin / 24 hodnot = 1 hodina na hodnotu). Největší skok je mezi 13:00 a 14:00 (1.36 kW). A protože jsme použili funkci `round()`, výstup je 1.36 (a ne 1.360). Tři nejvyšší hodnoty jsou 2.37 kW v 13:00, 1.42 kW v 08:00 a 1.39 kW v 11:00. Nejdelší stabilní interval je od 18:00 do 24:00 (6 hodin).

1.3 Příklad

Vstup

```
0.39,0.43,0.37,0.71,0.87,0.4,0.37,0.53,0.87,0.96,2.61,2.33,1.26,1.19,0.9,0.84,0.78,0.68,
0.43,0.44,0.45,0.37,-1.01,0.41
```

Výstup programu

```
ERROR
```

Komentář: Vstupní data obsahují minusovou hodnotu.

1.4 Příklad

Vstup

```
0.43,0.43,0.37,0.74,nan,0.33,0.45,1.31,0.78,1.18,1.2,1.41,1.44,1.04,1.21,0.97,0.75,0.75,
0.55,0.42,0.36,0.97,0.4,0.44
```

Výstup programu

ERROR

Komentář: Vstupní data obsahují hodnotu nan.