

## Lista 8

### Zadanie 1 (5 pkt)

```
if __name__ == "__main__":

    if len(sys.argv) != 3:
        usage()

    month, year = sys.argv[1:]

    day_of_the_week = get_day() # dzień tygodnia wybrany przez użytkownika

    # lista miesięcy rozpoczynających się wybranym dniem tygodnia
    # licząc od miesiąca i roku podanych z linii komend
    months = get_months(month, year, day_of_the_week)

    print_summary(months)
```

Uzupełnij powyższy program o brakujące funkcje tak, aby drukował listę miesięcy (w formacie MM-YYYY) - licząc od zadanego miesiąca z linii komend - które rozpoczęły się wybranym przez użytkownika dniem tygodnia. Przykładowo dla

```
month = 1
year = 2018
day_of_the_week = 1
```

program powinien drukować:

```
01-2018
10-2018
```

W przypadku podania złej liczby argumentów funkcja `usage` powinna drukować stosowny komunikat i przerwać działanie programu (*sys.exit*).

---

### Zadanie 2 (5 pkt)

Stwórz zmienną środowiskową `$LOG_PATH`, która wskazuje ścieżkę do katalogu, w którym zapisywane będą logi (np. `export LOG_PATH=$HOME/moje_logi`)

Napisz program, który:

- sprawdza, czy katalog wskazywany przez `$LOG_PATH` istnieje - jeśli nie istnieje, to go tworzy
- tworzy (jeśli nie istnieje) katalog `$LOG_PATH/YYYY-MM-DD`, gdzie YYYY - rok, MM - miesiąc, DD - dzień, w którym

- co  $N$  sekund tworzy plik `HH-MM-SS.log`, gdzie `HH` - godzina, `MM` - minuta, `SS` - sekunda, w którym
- umieszcza wartość zmiennej `$LOG_INFO`.

Napisz drugi program, który inicjuje `$LOG_INFO` wartością  $x$  i co  $n$  sekund zwiększa o ją o jeden.

$N$ ,  $n$  i  $x$  pobierane z linii komend (`sys.argv`).

Uruchom oba programy jednocześnie i zobacz, czy logi są generowane prawidłowo.

### Zadanie 3 (5 pkt)

Napisz program do gry w kółko i krzyżyk. Przykładowa plansza:

```

      1   2   3
A      |   |
-----
B    X |   |
-----
C      | O |

```