

# Wstęp do programowania 2016

## Pracownia 10

**Uwaga:** Na tej liście znowu będą wprawki o tematyce wybranej przez prowadzącego ćwiczenia. Podczas tych zajęć można oddawać zadania z listy 8 za 0.5 i 9 za 1.

Premia za tę listę wynosi 0.5, przyznawana jest osobom, które zdobyły co najmniej 1.5p za zadania z tej listy. Maksimum dla tej listy wynosi 3.5p.

**Zadanie 1.(1pkt)** Uruchom program sprawdzający spełnialność formuły ze strony wykładu. Bądź przygotowany na wyjaśnienie prowadzącemu wszystkich użytych w nim konstrukcji. Oprócz tego wykonaj następujące polecenia:

- Stwórz analogiczną funkcję, która sprawdza, czy formuła jest tautologią.
- Dodaj możliwość używania stałych w formule.
- Dodaj możliwość używania zmiennych o dłuższych nazwach (pisanych małymi literami)

Postaraj się, by po przeróbkach, Twój program pozostał możliwie zwięzły.

**Zadanie 2.(1pkt)** Łamigłówką arytmetyczną jest zadanie, w którym należy literom przyporządkować (różne) cyfry w ten sposób, by będące treścią zadania dodawanie było prawdziwe. Przykładowe zadania to:

SEND	CIACHO
+ MORE	+ CIACHO
-----	-----
MONEY	NADWAGA

Napisz program, który rozwiązuje łamigłówki arytmetyczne. W programie powinna być funkcja, której argumentem jest napis przedstawiający zagadkę (przykładowo "`send + more = money`", a wynikiem słownik kodujący (jakieś) rozwiązanie. Gdy rozwiązanie nie istnieje, funkcja powinna zwracać pusty słownik (ew. wartość `None`).

**Zadanie 3.(2pkt)** Napisz dwie funkcje wykorzystujące rekurencję (lub jedną za połowę punktów). W obu definicjach powinieneś skorzystać z mechanizmu *list comprehension*, postaraj się, by definicje były możliwie jak najbardziej zwięzłe.

- Napisz rekurencyjną funkcję, która generuje zbiór wszystkich sum podzbiorów listy liczb  $L$  (czyli jeżeli  $L$  była równa  $[1, 2, 3, 100]$ , to funkcja powinna zwrócić zbiór

```
set([0,1,2,3,4,5,6, 100, 101, 102, 103, 104, 105, 106])
```

- Napisz rekurencyjną funkcję, która generuje wszystkie ciągi niemalejące o długości  $N$ , zawierające liczby od  $A$  do  $B$ .

**Zadanie 4.(1pkt)** Jesteś pisarzem literatury fantastycznej (raczej użytkowej, niż artystycznej, szczerze mówiąc). Całkiem dobrze sobie z tym radzisz, ale masz kłopot z wymyślaniem imion dla bohaterów. To zadanie ma być użytecznym narzędziem rozwiązującym taki problem, czyli wspomagającym twórczy proces wymyślenia imion (nazwisk) dla tego typu literatury. Należy rozwiązać je w następujący sposób:

- Imię będziemy losować znak po znaku.
- Jak zobaczysz, wygodnie przyjąć, że każde imię zaczyna się od pary znaków ^^ a kończy znakiem \$ (oczywiście można tu wybrać inne oznaczenia, nie powinieneś również tych dziwnych znaków pokazywać użytkownikowi)
- Imię powinno mieć pewną długość minimalną, przykładowo 4 znaki.
- Prawdopodobieństwo wylosowania znaku na pozycji  $i$  powinno zależeć od znaków  $i-1$  oraz  $i-2$ .
- Prawdopodobieństwa te powinieneś szacować przeglądając plik z rzeczywistymi imionami (podany na kno, możesz skorzystać z innego – na przykład jednoznacznie słowiańskiego, jeżeli uda Ci się taki odnaleźć). Przykładowo, gdyby jedynymi imionami były Paweł i Ewelina, wówczas dla znaków `we` możliwe byłyby tylko dwie kontynuacje, mianowicie `ł` oraz `l`, każda z prawdopodobieństwem  $\frac{1}{2}$ .

Twój program powinien wczytać listę imion, oszacować na jej podstawie prawdopodobieństwo losowania znaków, następnie wylosować kilkanaście imion zgodnych z powyżej naszkicowanymi zasadami. Jeżeli nie do końca wiesz, jak się zabrać za to zadanie, śmiało pytaj prowadzącego na zajęciach lub na wykładzie.