

## Wstęp do programowania

### Pracownia 9

**Uwaga:** Na tej liście też będą wprawki (zalecany temat: rekurencja lub funkcja eval). Podczas tych zajęć można oddawać zadania z listy siódmej za 0.5 i ósmej za 1. Bonus (0.5) dla tej listy jest za zdobycie ponad dwóch punktów.

**Zadanie 1.(0.5pkt)** Wybierz i rozwiąż jedną wprawkę, której jeszcze nie robiłeś (może być z innej grupy), powinna mieć numer większy od 2.

**Zadanie 2.(1pkt)** Uruchom program sprawdzający spełnialność formuły ze strony wykładu 8. 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 3.(1pkt)** Wróćmy do zadania z przedstawianiem literek z imienia i nazwiska. Zmodyfikujemy je bardzo nieznacznie: z zadanych literek należy ułożyć nie dwa, lecz trzy słowa. Oznacza to w szczególności, że rozwiązanie, w którym znajdujemy słowa układalne i sprawdzamy wszystkie pary takich słów przestaje być akceptowalne (bo sprawdzenie wszystkich trójek trwa zbyt długo). Wskazówka: czy znając dwa słowa musisz przeglądać **wszystkie** układalne, żeby znaleźć trzecie? Wskazówka 2 (rot13): Cbzlfy b fybjavxh, j xgbelz xyhpmnzv fn cbfbegbjnar yvgrl quartb fybjn, n jnegbfpvn wrfg yvfgn fybj (wnxn?)

**Zadanie 4.(1pkt)** Pożegnianie z żółciem : wybierz rekurencyjny wzorek, którego wcześniej nie implementowałeś (możesz też znaleźć w Internecie inny rekurencyjny wzorek, taki jak paprotka z lodygą, drzewo Pitagorasa, dywan Sierpińskiego, albo zaproponować coś własnego).

Następnie napisz program używający modułu turtle, który ten rysunek wykonuje. Niezależnie od tego, jaki rysunek wybierzesz, możesz (lecz nie musisz) go uatrakcyjnić wizualnie (na przykład dodając kolory). Prowadzący może wynagrodzić premią (do 0.5) wyjątkowo atrakcyjne rozwiązania.

**Zadanie 5.(1.5+Xpkt)★** W zadaniu tym będziemy zajmować się Alfabetonami – czyli specjalnymi utworami z dowolnego rodzaju literackiego (proza, poezja, dramat), które składają się (oprócz znaków przestankowych i spacji) ze wszystkich liter polskiego alfabetu, przy czym każda występuje dokładnie raz.

Autorem pojęcia i wielu wibitnych przykładów<sup>1</sup> jest Stanisław Barańczak. Ułożenie alfabetonu jest dość trudne, więc zdefiniujemy też łatwiejszą jego wersję, tzn. półalfabeton, który zawiera różne litery, ale niekoniecznie wszystkie.

W zadaniu tym zajmiemy się opracowaniem narzędzi wspomagających trudną pracę alfabetonisty:

- Napisz program, który generuje półalfabeton, wybierając losowo kolejne słowa, tak aby cały czas spełniony był warunek unikalności liter. Program powinien kończyć działanie, gdy niemożliwe jest dołosowanie kolejnego słowa (postaraj się, by proces generowania był możliwie szybki), słowa powinieneś losować z pliku używanego w poprzednich listach. (0.8p)
- Wykonaj powyższy program wiele razy, sprawdzając, jakie litery za każdym razem pozostały i zliczając, w ilu przypadkach dana litera pozostała. Tę liczbę nazwiemy (szacowaną) trudnością litery. (0.5)
- Posortuj słowa z niepowtarzającymi się literami ze względu na średnią trudność liter w nich występujących. Wybierz jedno słowo, które wydaje ci się ciekawe, a jednocześnie zawiera trudne litery. Zmodyfikuj generator z a), by mógł startować z ułożoną częścią alfabetonu i uruchom go dla tego słowa. (0.3)

Z zadaniem jest związany Wielki Konkurs Alfabetoniczny, opisany na SKOS.

---

<sup>1</sup> Choćby: Chrztań ufną belgijskość, tęp wyż-młódź!