

## Wstęp do programowania 2013

### Pracownia 6

**Uwaga:** Na tej liście też będą wprawki (zalecany temat: funkcja licząca coś na listach). Podczas tych zajęć można oddawać zadania z listy czwartej za 0.5 i późniejszych za 1.

Premia za tę listę wynosi 0.5, wlicza się do maksimum, przyznawana jest osobom, które zdobyły co najmniej 2p za zadania z tej listy.

**Zadanie 1.(1pkt)** Na stronie wykładu znajdziesz link prowadzący do opisu Maszyny RAM (być może znasz ją ze Wstępu do informatyki). Twoim zadaniem będzie napisanie interpretera naszej wersji maszyny RAM, wykorzystując umieszczony na kno program `ram_niepelny.py` (jeżeli chcesz, możesz nie korzystać z tego programu). W celu ułatwienia implementacji, nasza wersja będzie się nieznacznie różniła od wersji w WdI (co jest dokładnie opisane w osobnym dokumencie).

**Zadanie 2.(1pkt)** Parę wyrazów nazwiemy wzajemnie odwrotnymi, jeżeli pierwszy z nich jest równy drugiemu przeczytanemu wspak. Przykładowo: *zakup* oraz *pokaz*. Na stronie wykładu jest plik z polskimi słowami, Twoim zadaniem jest napisać program, który wypisuje wszystkie wzajemnie odwrotne pary słów. Każda para powinna być wypisana raz (czyli jeżeli wypisałeś parę zakup-pokaz, to nie powinieneś wypisywać pary pokaz-zakup). Uwaga: program powinien działać szybko, zastanów się jak uniknąć pętli w pętli (do generowania wszystkich par słów).

**Zadanie 3.(1pkt)** Mówimy, że jakieś słowo jest układalne z drugiego, jeżeli da się z tego drugiego wybrać pewne (może wszystkie) literki i ułożyć to pierwsze (być może zmieniając kolejność). Przykładowo *aktyw*, *kot*, *motyl* są układalne z *lokomotywa*, a *żak* i *kotka* – nie są (bo lokomotywa nie ma w sobie *ż* i ma tylko jedno *k*).

Napisz funkcję, która sprawdza, że jakieś słowo jest układalne z drugiego. Twoja funkcja powinna wykorzystywać inną funkcję, która zamienia słowo *s* na słownik, mówiący ile razy w *s* występuje dana literka.

(zadanie to będzie miało kontynuację na dalszych listach, zatem warto je zrobić)

**Zadanie 4.(1pkt)** Napisz program, który rozwiązuje następujące łamigłówki: wczytuje z pliku  $2 \times K$  słów, łączy je w  $K$  par w taki sposób, żeby każda para składała się ze słowa wewnętrznego (*a*) oraz zewnętrznego (*b*), i żeby słowo zewnętrzne dało się podzielić na dwie części ( $b_1, b_2$ ) w taki sposób, że po połączeniu z *a* dadzą istniejące słowo, mające postać  $b_1ab_2$  (zarówno  $b_1$  jak i  $b_2$  powinny być niepuste). Przykładowo, warunki te spełnia para (*łom*, *pień*) bowiem istnieje podział słowa *pień* na część *p*- oraz *-ień*, w wyniku czego otrzymamy słowo *płomień*.

Zaprezentuj działanie programu na następującym zestawie słów: kotka, kula, kutwa, lin, matura, para, pranie, ropa, setka, sowa.