

Wstęp do programowania 2017

Pracownia 3

Podczas tych zajęć można oddawać zadania z listy pierwszej i drugiej za 1 punkt. Na trzecich zajęciach jeszcze nie ma wprawek, ale na kolejnych **już będą**. Premia dla tej listy wynosi 0.5 za rozwiązanie co najmniej 2 zadań. Premia wlicza się do maksimum.

Na stronie wykładu znajdują się rysunki, stanowiące ilustracje do zadania ostatniego z tej listy. Używane są tam następujące oznaczenia: Ł - zadanie jest łatwe, L - w zadaniu należy skorzystać z generatora liczb pseudolosowych (zatem Twój rysunek może się nieco różnić, bo wylosujesz inne liczby), R - wygodnie jest użyć rekurencji, T - zadanie wydaje się nieco trudniejsze. O rekurencji będziemy jeszcze mówić na wykładzie, zatem nie przejmuj się, jeżeli nie wiesz, jak robić zadania z literką R. O bibliotece turtle powiemy sobie trochę na trzecim wykładzie.

Zadanie 1.(1pkt) Liczba pierwsza to taka liczba większa niż 1, która bez reszty dzieli się tylko przez 1 i przez samą siebie. Liczbę nazwiemy szczęśliwą, jeżeli jej zapis dziesiętny zawiera 3 siódemki pod rząd. Napisz program, który wypisuje wszystkie szczęśliwe liczby pierwsze z zakresu od 1 do 100000 wraz z informacją, ile takich liczb jest. Program powinien zawierać funkcję, która sprawdza, czy dana liczba jest pierwsza. Wskazówka: sprawdź, co oznacza w Pythonie `napis in inny_napis`. Wskazówka 2: wystarczy proste rozwiązanie sprawdzające wszystkie liczby i wypisujące te z nich, które spełniają warunki.

Zadanie 2.(1pkt) Liczba pierwsza jest hiperszczęśliwa, jeżeli zawiera co najmniej 7 siódemek pod rząd. Napisz program, który wypisuje największą dziesiętocyfrową hiperszczęśliwą liczbę pierwszą. Postaraj się, by Twój program można było łatwo zastosować w innym, po dobnym zadaniu (przy innej liczbie cyfr i innej liczbie siódemek czyniących liczbę hiperszczęśliwą). Program powinien zakończyć działanie w kilkanaście sekund.

Zadanie 3.(1pkt) Napisz funkcję `usun_nawiasy(s)`, która bierze napis i usuwa z niego fragmenty w nawiasach (okrągłych). Czyli na przykład `usun_nawiasy("Ala ma kota (perskiego)!")` powinno dać napis `'Ala ma kota!'`. Nawiasów może być więcej niż jeden, ale możesz założyć, że się nie zagnieżdżają i są prawidłowo rozłożone (czyli występują parami, najpierw otwierający, potem zamykający). Dołącz do funkcji kilka jej testowych wywołań.

Na trzecim wykładzie powinno być powiedziane o:

- a) jak przeglądać napisy,
- b) jak zmieniać listy na napisy i odwrotnie.

Zadanie 4.(1pkt) Wybierz jeden rysunek, a następnie używając biblioteki turtle stwórz program, który rysuje taki sam lub bardzo podobny rysunek. Uwaga: punktacja nie zależy od trudności rysunku. Rysunki powinny składać się z czarnych kresek (kolorowe literki nie są częścią rysunku i nie trzeba ich rysować).