

Wstęp do programowania 2012

Pracownia 1

Dla każdego z poniższych zadań powinieneś napisać program, który umieścisz w osobnym pliku. Programy te nie będą pobierały żadnych danych, będą natomiast wypisywać na standardowym wyjściu pewne teksty. Ponieważ to pierwsza lista, zarówno w terminie +1 jak i w terminie +2 nie obowiązują kary za spóźnione oddawanie. Na tych zajęciach jeszcze nie ma wprawek. Przysługuje dodatkowa premia (+0.5) za tę listę, jeżeli oddasz jakiś program na pierwszych zajęciach.

Zadanie 1.(1pkt) Uruchom program, który na stronie kno jest przypisany pierwszego wykładu. Postaraj się zrozumieć wszystkie użyte w nim konstrukcje (jeżeli coś jest niejasne, to nie krępuj się spytać prowadzącego pracownię). Następnie zmodyfikuj ten program, by wypisywał tekst umieszczony na stronie wykładu i by nie zawierał niepotrzebnych elementów.

Zadanie 2.(1pkt) Jak wiadomo $n! = 1 \cdot 2 \cdot \dots \cdot n$. Napisz w pythonie definicję funkcji silnia. Napisz program, który wypisuje następujący raport:

```
4! ma 2 cyfry
5! ma 3 cyfry
6! ma 3 cyfry
7! ma 4 cyfry
8! ma 5 cyfr
...
```

Raport powinien kończyć się na liczbie 100. Przy liczeniu liczby cyfr możesz skorzystać z następujących funkcji (nie opisujemy ich działania, jedynie podajemy przykładowe wywołania. Spróbuj, znając je i wpisując w interpreterze, domyśleć się tego działania:

```
str(123)
str(123)+str(123)
int(3.141592)
len("ala ma kota")
log(10)
log(2.7182818284590451)
log(1000)/log(10)
```

Funkcja `log` musi być zaimportowana z modułu `math`. Liczbę cyfr można liczyć na dwa sposoby. Wybierz jeden, a prowadzącemu opowiedz o drugim.

Zadanie 3.(1pkt) Jesteś w jaskini, której wyjścia pilnuje smok. Smok wyjmie kostkę (na szczęście taką sześcienną, do gry) i proponuje następującą grę: rzucasz sto razy tą kostką, jeżeli w tych stu rzutach jakaś liczba wypadnie co najmniej 6 razy pod rząd, wówczas możesz wyjść. W przeciwnym przypadku musisz rzucać jeszcze raz. Napisz program, który symuluje tę grę, czyli rzuca 100 razy „pseudokością”, wypisuje informację o tym, ile elementów ma najdłuższy podciąg takich samych elementów i (jeżeli jest ich co najmniej 6) przerywa grę, informując o tym, ile sturzutowych serii się odbyło.

Mogą ci się przydać dwie informacje dotyczące Pythona:

- Następująca pętla wykonuje potencjalnie nieskończone obliczenia, a w momencie, gdy w instrukcji `if` spełniony będzie warunek, kończy swoje działanie:

```
while True:
    ...
    if WarunekKonca:
        break
    ...
```

- Losowanie można wykonywać za pomocą funkcji `randint` z modułu `random`. Czyli program:

```
from random import randint
print randint(0,9)
print randint(0,9)
print randint(0,9)
print randint(0,9)
```

wypisze cztery cyfry, wśród których może znaleźć się zarówno 0 jak i 9

Zadanie 4.(1pkt) Hasła losowane są dość bezpieczne, niemniej trudno się je zapamiętuje. Twoim zadaniem będzie napisać funkcję, która losuje łatwe do zapamiętania hasło (które zarazem pozostaje dość bezpieczne). Takie hasło będzie się składać z losowych krótkich fragmentów tekstu, tak dobranych, że łatwo się je wymawia po połączeniu. Funkcja losująca te fragmenty znajduje się w module `losowanie_fragmentow`, który powinieneś pobrać ze strony wykładu, umieścić w kartotece, w której znajduje się Twój program. Wówczas program:

```
from losowanie_fragmentow import losujFragment
for i in range(5):
    print losujFragment()
```

może wypisać coś w stylu:

```
sia
je
tra
kawa
ce
```

(wiemy, że istnieją fragmenty o długości 2,3 oraz 4)

Musisz napisać funkcję `losujHaslo(n)`, która losuje hasło, składające się z wspomnianych fragmentów, o długości dokładnie `n` znaków. Zadeemonstruj działanie tej funkcji pisząc program, który wywołuje ją 10 razy dla `n=8` oraz 10 razy dla `n=12` i wypisuje wyniki. Uwaga: każdy fragment powinien mieć możliwość zostania wylosowanym.