

# Wstęp do programowania 2020

## Pracownia 1

Data publikacji: 3.10.2020

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<sup>1</sup>. Na tych zajęciach jeszcze nie ma wprawek. Przysługuje dodatkowa premia (+0.5) za tę listę, jeżeli oddasz jakiś program najpóźniej na pierwszych zajęciach po terminie wykładu.

**Zadanie 1.(1pkt)** Uruchom program, który na stronie **skos** 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, domysleć się tego działania, większość zresztą była omawiana na wykładzie):

```
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. **Uwaga:** napisy powinny być poprawne gramatycznie (czyli musisz użyć właściwej formy słowa *cyfra*). Jeżeli tego nie zrobisz, za zadanie dostaniesz 0.8p.

**Zadanie 3.(0.5pkt)** Napisz procedurę `krzyzyk(n)`, która rysuje krzyżyk z gwiazdek, składający się z 5-ciu kwadratów o boku `n`, jak na poniższym rysunku (na którym pokazany jest wynik wywołania `krzyzyk(4)`). „Rysowanie” jest oczywiście wykonywane za pomocą wywołania funkcji `print`.

```
****
****
****
****
*****
*****
*****
*****
```

---

<sup>1</sup>Na stronie wykładu będzie wyjaśnienie, jaki dokładnie rozumieć +1 i +2. Przybliżone rozumienie: +k oznacza k tygodni od publikacji, ułamki zaokrąglane są w dół

\*\*\*\*  
\*\*\*\*  
\*\*\*\*  
\*\*\*\*

**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_fagmentow`, który powinieneś pobrać ze strony wykładu, umieścić w kartotece, w której znajduje się Twój program. Wówczas program:

```
from losowanie_fagmentow import losuj_fragment
for i in range(5):
    print (losuj_fragment())
```

może wypisać coś w stylu:

```
sia
je
tra
kawa
ce
```

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

Musisz napisać funkcję `losuj_haslo(n)`, która losuje hasło, składające się z wspomnianych fragmentów, o długości dokładnie `n` znaków. Zademonstruj 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.