

- Rozwiązane zadania należy wysłać mailem do prowadzącego nie później niż w ciągu 90 minut licząc od rozpoczęcia kolokwium.
 - Każde zadanie należy umieścić w osobnym pliku `imie.nazwisko.kolokwium.XX.py` gdzie `XX` to numer zadania.
 - Punktacja:
 - bdb: ≥ 5.0 pkt
 - db+: ≥ 4.5 pkt
 - db: ≥ 4.0 pkt
 - dst+: ≥ 3.5 pkt
 - dst: ≥ 3.0 pkt
-

Zadanie 1 (2 pkt)

Napisz program, który rysuje trójkąt na podstawie podanego przez użytkownika (jako argument wywołania) tekstu wg poniższego wzoru (wygenerowanego dla “kolokwium”):

```
k
ko
kol
kolo
kolok
kolokw
kolokwi
kolokwiu
kolokwium
kolokwiu
kolokwi
kolokw
kolok
kolo
kol
ko
k
```

Zadanie 2 (2 pkt)

Napisz program, który generuje liczbę 4-cyfrową a następnie prosi użytkownika o jej odgadnięcie (pobierane ze standardowego wejścia). Użytkownik zgaduje do

skutku. Po każdej próbie użytkownik dostaje wskazówkę, o liczbie cyfr, które odgadł:

- 10 pkt za każdą cyfrę na dobrej pozycji;
- 1 pkt za każdą cyfrę na złej pozycji.

Np.

```
losowa_liczba = 1234
```

```
8765 -> 0
```

```
9041 -> 2
```

```
1423 -> 13
```

```
...
```

Uwaga: program powinien stosownie reagować, gdy podane przez użytkownika dane są niepoprawne.

Zadanie 3 (2 pkt)

Napisz program, który tworzy plik `score.txt`, w którym umieszcza kolejne litery alfabetu wraz z losową cyfrą (liczba punktów za daną literę), oddzielone średnikiem, np.

```
a 3; b 2; c 5...
```

Następnie:

- wczytuje plik `score.txt` i na podstawie jego zawartości tworzy słownik punktacji, np.

```
punktacja = {'a': 3, 'b': 2, 'c': 5 ...
```

- prosi użytkownika o podanie słowa;
- informuje o liczbie zdobytych punktów;
- zwraca stosowny komunikat i przerywa działanie programu w przypadku podania niepoprawnych danych (np. pusty string, cyfry, więcej słów itp.).