

Kurs języka Python

Lista 1.

Zadanie 1. Zaprogramuj funkcję `rzut_kostka()`, która symuluje rzut sześcienną kością do gry, tzn. funkcja ma zwracać losową liczbę naturalną z przedziału $[1 \dots 6]$. Korzystając z tej funkcji zaprogramuj prostą grę (z dwoma zawodnikami), w której w każdej turze każdy zawodnik rzuca dwie kostki (tj. dwukrotnie wywołuje funkcję `rzut_kostka()`). Turę wygrywa ten, kto wyrzuci więcej kostek. Grę wygrywa ten, kto po n turach ma więcej zwycięstw. Jeśli po n turach jest remis, gra toczy się do pierwszego zwycięstwa. Komputer ma grać sam ze sobą, a dodatkowo po każdej turze wypisywać na wylosowaną liczbę oczek oraz aktualny bilans zwycięstw i porażek. Liczba n jest parametrem wywołania programu.

Zadanie 2. Napisz program który szyfruje tekst za pomocą następującego algorytmu opartego na algorytmie XOR: do zaszyfrowania jest potrzebny klucz k , tj. liczba z przedziału $[0 \dots 255]$. Kolejne litery tekstu zamieniamy na odpowiedni kod ASCII, obliczamy wynik operacji XOR z k i do szyfrogramu wstawiamy wynik operacji zamieniony na odpowiedni znak ASCII. Na przykład tekst *Python* za pomocą klucza 7 (binarnie: 0000 0111) szyfrujemy tak:

litery	P	y	t	h	o	n
ASCII	0101 0000	0111 1001	0111 0100	0110 1000	0110 1111	0110 1110
XOR	0101 0111	0111 1110	0111 0011	0110 1111	0110 1000	0110 1001
szyfr	W	~	s	o	h	i

Program ma mieć postać funkcji `zaszyfruj(tekst, klucz)`, która dla podanego tekstu i klucza zwraca zaszyfrowany tekst. Zaprogramuj również funkcję `odszyfruj(szyfr, klucz)`.

Zadanie 3. Zaprogramuj funkcję `sloownie(n)`, która zwraca string będący słownym zapisem liczby naturalnej n , zapisanej zgodnie z regułami języka polskiego. Można przyjąć, że argumentem jest liczba o co najwyżej 8 cyfrach.

Zadanie 4. Napisz jednoargumentową funkcję `rozklad(n)` która oblicza rozkład liczby n na czynniki pierwsze i zwraca jako wynik listę par $[(p_1, w_1), (p_2, w_2), \dots, (p_k, w_k)]$ taką, że

$n = p_1^{w_1} * p_2^{w_2} * \dots * p_k^{w_k}$ oraz p_1, \dots, p_k są różnymi liczbami pierwszymi. Na przykład

```
>>> rozklad(756)
[(2, 2), (3, 3), (7, 1)]
```

Zadanie 5. Zaprogramuj funkcję `tabliczka(x1, x2, y1, y2)`, która wypisze na ekran tabliczkę mnożenia dla liczb $[x_1, \dots, x_2] \times [y_1, \dots, y_2]$; np.

```
>>> tabliczka(3,5, 2, 4)
  3  4  5
2 6  8 10
3 9 12 15
4 12 16 20
```

Możesz zaniedbać formatowanie (wyrównywanie kolumn).

Każde zadanie jest warte 2 punkty. Na pracowni do oceny należy przedstawić dwa zadania.