

Wstęp do programowania. Lista 5. Pętla *for*.

1. Napisz program, który wyświetli na ekranie 100 znaków \* na dwa sposoby (używając pętli *for* oraz tylko polecenia *print*).
2. Napisz program, który wyświetli wszystkie liczby naturalne mniejsze równe 150.
3. Napisz program, który po wczytaniu łańcucha wyświetli go "pionowo". Użyj pętli *for*. (Na przykład po wczytaniu *wyraz* wyświetli:  
*w*  
*y*  
*r*  
*a*  
*z*.)
4. Napisz program, który po wczytaniu łańcucha wyświetli go bez znaków interpunkcyjnych. (sprawdź *punctuation* w bibliotece *string*).
5. Napisz program, który sprawdza czy elementy krotki  $t = (-5, 16, 3.6, 123, -89)$  są mniejsze bądź równe od liczby podanej przez użytkownika.
6. Napisz program, który utworzy i wyświetli listę złożoną z 10 losowych liczb całkowitych z przedziału  $\langle -2, 18 \rangle$ . (patrz biblioteka *random*)
7. Napisz program, który utworzy i wyświetli listę złożoną z 10 losowych liczb całkowitych z przedziału  $\langle -2, 18 \rangle$ , następnie stworzy listę składającą się z tych elementów pierwszej listy, które są mniejsze od 10. (Dwukrotnie zastosuj listę składaną).
8. Napisz program, który stworzy i wyświetli słownik składający się z elementów *'litera':kwadrat jej pozycji w alfabecie* (tylko małe litery, bez polskich znaków). Gdzie wielkość słownika określa użytkownik (od 1 do 26).  $\{ 'a' : 1, 'b' : 4, \dots \}$  (sprawdź funkcje *chr*, *ord*).
9. Napisz program, który po wczytaniu łańcucha znaków z klawiatury wyświetli słownik postaci  $\{ 'znak':ilość wystąpienia znaku, \dots \}$ .
10. Napisz program, który poprosi użytkownika o podanie hasła dostępu. W przypadku podania prawidłowego hasła program ma zakończyć działanie, w przypadku błędnego hasła program ma ponowić prośbę i podać ile prób pozostało użytkownikowi. Maksymalna ilość prób to 5.
11. Napisz program, który wyświetli

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

12. Napisz program, który obliczy i wyświetli ile jest liczb parzystych i nieparzystych w krotce (4, 6, 2, 43, 112, 344, 234, 678, 999, 44, 1, 97).
13. Napisz program, który policzy ile kwadratów liczb całkowitych od 1 do 1000 ma jako ostatnią cyfrę 4, a ile 9.
14. Napisz program, który wyświetli tabliczkę mnożenia do 100. Zadbaj o wizualną prezentację.
15. Napisz program, który wyświetli listę wszystkich liczb pierwszych do 100. (zastosuj *break*).
16. Napisz program, który wyświetli ile jest liczb pierwszych w przedziale (100, 1000).
17. Napisz program, który wyświetli tablicę wymiaru  $5 \times 8$  wypełnioną losowymi liczbami całkowitymi z przedziału  $\langle 5, 9 \rangle$ . Następnie wyświetli ją przemieszaną (i wiersze i elementy w wierszach).
18. Napisz program, który wyświetli tablicę wymiaru  $5 \times 8$  wypełnioną losowymi liczbami całkowitymi z przedziału  $\langle 5, 9 \rangle$  (patrz biblioteka *random*). Następnie wyświetli listę oraz policzy ile jest w niej liczb nieparzystych i wyświetli odpowiedź. Postaraj się użyć list składanych.
19. Napisz program, który szyfruje podany tekst kodem Cezara, oraz odszyfrowuje podany tekst z zadaniem przez użytkownika przesunięciem, wyświetlając wyniki. Pracuj tylko na małych literach i bez polskich znaków. (Szyfr Cezara–jest to szyfr zbudowany na zasadzie szyfru podstawieniowego gdzie każda litera, tekstu niezaszyfrowanego zastępowana jest oddaloną od niej o stałą liczbę pozycji w alfabecie inną literą.)
20. Napisz program, który rozwiąże za ciebie zagadkę: w misce jest mniej niż 200 cukierków, wiesz, że dzieląc cukierki po równo na 5 osób zostaną 2 cukierki, dzieląc równo na 6 osób zostaną 3 cukierki, a dzieląc na 7 osób zostaną 2 cukierki. Ile cukierków jest w misce? Czy jest kilka możliwości?
21. Napisz grę/zabawę sprawdzającą znajomość tabliczki mnożenia. Zabawa ma mieć 10 rund. Każda runda to pytanie o wynik mnożenia w zakresie do 100, a następnie program ma informować czy to dobry czy zły wynik, jeśli zły to jaki jest dobry. Na koniec program ma podać ilość dobrych i złych odpowiedzi. Pogratulować, gdy wszystkie odpowiedzi były dobre.
22. Napisz program wyświetlający tyle wierszy trójkąta Pascala, ile zechce użytkownik. Zadbaj o formatowanie (przynajmniej gdy użytkownik poda do 10 wierszy, sprawdź polecenie `print(*l, sep='')` gdzie *l* lista):

```

                1
              1  1
             1  2  1
            1  3  3  1
           1  4  6  4  1

```