

## Lab 4

### Pętla for

for zmienna in range(wartość1, wartość2, wartość3):  
instrukcje do wykonania w pętli

wartość1 - początkowa wartość zmiennej (domyślnie: 0).

wartość2 - końcowa wartość zmiennej, która nie zostanie przekroczona.

wartość3 - wartość określająca o ile zwiększona ma być wartość zmiennej (domyślnie: 1).

## Zadania

1. Napisz program, który pobierze od użytkownika liczbę  $n$ . Następnie będzie pobierał  $n$  liczb oraz wyświetli ich średnią.

Wejście

Zmienna  $n$  przechowująca liczbę liczb do pobrania oraz zmienna liczba, przechowująca kolejne podane liczby.

Wyjście

Na ekranie pojawia się średnia pobranych liczb.

Warunki poprawności zadania

Po podaniu wszystkich liczb wyświetla się ich średnia. Upewnij się, że prawidłowa wartość pojawia się przy podaniu liczb ujemnych, ułamkowych.

2. Napisz program, który pobierze od użytkownika liczbę  $n$ , następnie wyświetli liczby dodatnie, które są podzielne przez 4 oraz mniejsze od  $n$ .

Wejście

Zmienna  $n$  przechowująca pobraną liczbę całkowitą dodatnią.

Wyjście

Na ekranie pojawiają się wszystkie liczby, które są mniejsze od pobranej liczby oraz podzielne przez 4.

Warunki poprawności zadania

Po podaniu liczby całkowitej dodatniej pojawiają się wszystkie liczby całkowite, które są mniejsze od pobranej liczby oraz podzielne przez 4.

3. Napisz program, który pobierze od użytkownika liczbę  $n$ . Następnie wyświetli informację jaka jest wartość liczby  $n!$ .

Wejście

Zmienna  $n$  przechowująca pobraną liczbę całkowitą dodatnią.

Wyjście

Na ekranie pojawia się silnia pobranej liczby.

Warunki poprawności zadania

Po podaniu liczby całkowitej dodatniej wyświetla się silnia.

4. Napisz program, który będzie pobierał od użytkownika liczbę  $n$ , a następnie wyświetli 2 "trójkąty prostokątne" skierowane w lewą stronę od góry i dołu z gwiazdek o wysokości podanej przez użytkownika. Przykład dla  $n$  równego 5, gdzie znak - to puste miejsce, znak spacji " ":



#### Wejście

Zmienna  $n$  przechowująca dodatnią liczbę całkowitą, będące wysokością trójkąta.

#### Wyjście

Na ekranie pojawiają się 2 "trójkąty prostokątne" skierowane w lewą stronę od góry i dołu z gwiazdek o podanej wysokości przez użytkownika.

#### Warunki poprawności zadania

Po podaniu wysokości przez użytkownika pojawiają się 2 "trójkąty prostokątne" skierowane w lewą stronę od góry i dołu z gwiazdek o podanej wysokości przez użytkownika.

5. Napisz program, który będzie pobierał od użytkownika liczbę  $n$ , a następnie wyświetli 2 "trójkąty prostokątne" skierowane w prawą stronę od góry i dołu z gwiazdek o podanej wysokości przez użytkownika. Przykład dla  $n$  równego 5:

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

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

#### Wejście

Zmienna  $n$  przechowująca dodatnią liczbę całkowitą, będące wysokością trójkąta.

#### Wyjście

Na ekranie pojawiają się 2 "trójkąty prostokątne" skierowane w prawą stronę od góry i dołu z gwiazdek o podanej wysokości przez użytkownika.

Warunki poprawności zadania

Po podaniu wysokości przez użytkownika pojawiają się 2 "trójkąty prostokątne" skierowane w prawą stronę od góry i dołu z gwiazdek o podanej wysokości przez użytkownika.

6. Napisz program, który będzie pobierał od użytkownika liczbę  $n$ , a następnie wyświetli "choinkę" z gwiazdek o wysokości podanej przez użytkownika. Przykład dla  $n$  równego 5:

```

      *
     ***
    *****
   *********
  ***********
 |_ |
```

Wejście

Zmienna  $n$  przechowująca dodatnią liczbę całkowitą, będące wysokością "choinki".

Wyjście Na ekranie pojawia się "choinki" z gwiazdek wraz z "korzeniem" na środku.

Warunki poprawności zadania Po podaniu wysokości przez użytkownika pojawia się "choinka" z gwiazdek wraz z korzeniem. Upewnij się, że odpowiednia, że po podaniu liczby pojawia się "choinka" o odpowiedniej długości wraz z "korzeniem" na środku.

7. Napisz skrypt, który pobierze od użytkownika liczbę  $n$ , a następnie wyświetli wszystkie kombinacje umieszczenia liczb z zakresu  $1...n$  w pięciu pudełkach w taki sposób, że liczby mogą się powtarzać. Przykład dla  $n$  równego 3:

```

1 1 1 1 1
1 1 1 1 2
1 1 1 1 3
1 1 1 2 1
1 1 1 2 2
1 1 1 2 3
1 1 1 3 1
...
3 3 3 3 1
3 3 3 3 2
3 3 3 3 3
```

Wejście

Zmienna  $n$  przechowująca dodatnią liczbę całkowitą.

Wyjście

Na ekranie pojawia się wszystkie kombinacje umieszczenia liczb całkowitych z zakresu  $1...n$ .

Warunki poprawności zadania

Po podaniu liczb całkowitej przez użytkownika pojawiają się wszystkie kombinacje rozmieszczenia

liczb całkowitych z zakresu od 1 do  $n$  w taki sposób, że jedna liczba może zostać umieszczona w kilku pudełkach.

8. Napisz program, który pobierze od użytkownika liczbę  $n$ , a następnie wypisze  $n$ -tą liczbę Fibonacciego.

Wejście

Zmienna  $n$ .

Wyjście

Na ekranie pojawia się  $n$ -ta liczba Fibonacciego.

Warunki poprawności zadania

Upewnij się, że na ekranie pokazuje się prawidłowa liczba.

9. Napisz program, który pobierze od użytkownika hasło oraz liczbę  $n$ . Następnie poprosi użytkownika o podanie prawidłowego hasła  $n$  razy. W przypadku podania nieprawidłowego hasła, użytkownik powinien wpisywać je tak długo, aż poda prawidłowe. Przykład działania programu:

Podaj hasło: kot

Podaj ilość powtórzeń: 3

Powtorz hasło 1: kot123

Powtorz hasło 1: kot

Powtorz hasło 2: kotek

Powtorz hasło 2: kocurek

Powtorz hasło 2: kot

Powtorz hasło 3: kot

Dziękuję, podałeś prawidłowe hasło 3 razy

Wejście

Zmienna  $n$  przechowująca liczbę całkowitą, będącą liczbą powtórzeń oraz zmienna hasło przechowująca podane hasło.

Wyjście

Na ekranie pojawia się komunikat informujący użytkownika, że podał poprawne hasło  $n$  razy.

Warunki poprawności zadania

Po podaniu prawidłowego hasła na ekranie powinien pojawić się odpowiedni komunikat. Upewnij się, że po wpisaniu nieprawidłowego hasła, program poprosi użytkownika o podanie prawidłowego hasła kolejny raz oraz, że podane hasło zostanie ponownie wpisane  $n$  razy.