

■ Rekurencja

```
static long Silnia(int n)
{
    if (n == 1)
    {
        return 1;
    }

    return n * Silnia(n - 1);
}
```

Zasada działania rekurencji:

Zadanie 6.1 Napisać program do rekurencyjnego obliczania n-tego wyrazu ciągu Fibonacciego. Wskazówka: Ciąg Fibonacciego definiuje się następująco:

$$a_0 = 1, a_1 = 1, \dots, a_n = a_{n-1} + a_{n-2}$$

Zadanie 6.2 Napisać program do poszukiwania drogi wyjścia z labiryntu. Program przetestować na następującym labiryncie:¹

```
# # # # # # # # # # #
# . . . # . . . . . #
. . # . # . # # # . #
# # # . # . . . # . #
# . . . . # # # . # .
# # # # . # . # . # .
# . . # . # . # . # .
# # . # . # . # . # .
# . . . . . . . # . #
# # # # # # . # # # . #
# . . . . . # . . . #
# # # # # # # # # # #
```

Algorytm: „Kładziemy prawą rękę na ścianie i posuwamy się do przodu”. Jeśli labirynt ma inne wyjście, algorytm gwarantuje znalezienie tego wyjścia. Jeśli labirynt nie ma drugiego wyjścia, wrócimy do punktu startowego. Algorytm nie gwarantuje najkrótszej drogi.

Wskazówka: Labirynt umieścić w tablicy dwuwymiarowej typu char. Ściany labiryntu oznaczamy znakiem '#', a korytarze – kropką. Każdy wykonany ruch zaznaczamy znakiem 'x'.

¹ Zadanie pochodzi z książki Deitela