Celem ćwiczenia jest zapoznanie się z pętlami, szczególnie `for`, podstawowymi typami liczbowymi (`int, double`) oraz instrukcją warunkową `if`.

1.1 Programy `suma` i `silnia`

Napisać program, który z użyciem pętli `for` oblicza sumę

$$s = \sum_{i=1}^{n} i = 1 + 2 + 3 + 4 + \ldots + n$$

oraz iloczyn

$$s = \prod_{i=1}^{n} i = 1 \times 2 \times 3 \times 4 \times \ldots \times n$$

kolejnych liczb *naturalnych*. Sprawdzić wynik za pomocą znanych wzorów i wartości dla powyższych wyrażeń.

UWAGA: *dla dużych wartości* $n \gg 10$ *powyższe wyrażenia, szczególnie* $n!$*, mogą przekroczyć zakres zmiennej* $\int$*. Proszę w drugim podejściu spróbować rozszerzyć program tak aby odpowiednio reagował na przekroczenie zakresu, np: wypisując komunikat o błędzie i przerywając obliczenia.*

1.2 Program obliczający podwójną silnię $n!!$

Zmodyfikować program z poprzedniego zadania tak, aby obliczał iloczyn co drugiej liczby całkowitej

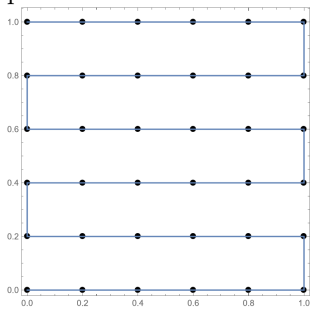
$$n!! = n \times (n-2) \times (n-4) \times \ldots$$

do momentu aż czynniki powyższego iloczynu pozostaną dodatnie. Wynik powinien być typu `double`.

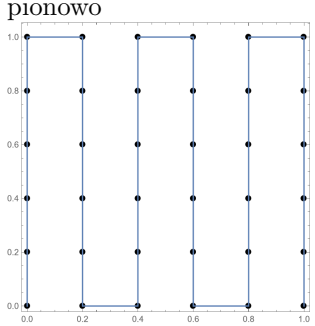
1.3 Krzywa wypełniająca kwadrat.

Napisać program **fill**, przyjmujący na wejściu zbiór $N \times N$ punktów równomiernie rozłożonych w kwadracie $[0,1] \times [0,1]$ (z włączeniem brzegów), a na terminalu wypisujący N^2 punktów w kolejności:

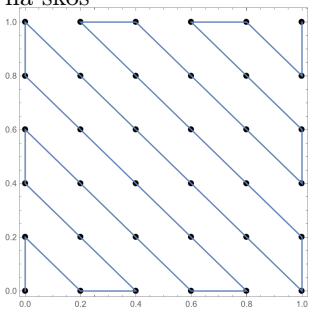
1. poziomo



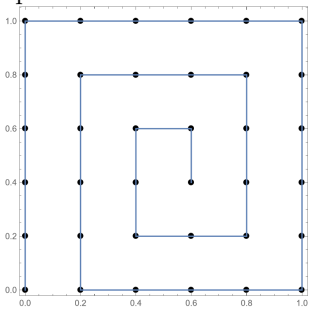
2. pionowo



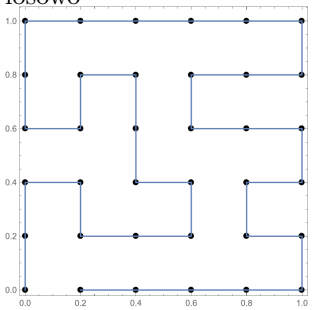
3. na skos



4. spiralnie



5. losowo



6. z użyciem *Z-order curve*

7. krzywej Hilberta (dla $N = 2^n$)
8. krzywej Peano (dla $N = 3^n$)
9. z użyciem własnego pomysłu

Skanowanie zaczyna się od punktu $\{0, 0\}$. Przykładowy efekt uruchomienia programu. Formatowanie wyniku pozwala na dołączenie go jako kodu w języku C.

```
./fill 3
{
{0., 0.},
{0.5, 0.},
{1., 0.},
{1., 0.5},
{0.5, 0.5},
{0., 0.5},
{0., 1.},
{0.5, 1.},
{1., 1.}
};
```

UWAGA: *idea programu polega na przeskanowaniu przestrzeni 2-wymiarowej (parametrowej) w taki sposób, aby odwiedzić każdy punkt na płaszczyźnie jeden raz, ale bez dokonywania „dużych" przeskoków i przy zachowaniu „lokalności". Przez lokalność rozumiemy takie uporządkowanie punktów, że punkty sąsiadujące ze sobą na liście jednowymiarowej są także bliskie na wejściowej płaszczyźnie 2-wymiarowej.*