Trójkątów ciąg dalszy. Tym razem wprawka składa się z dwóch części każda za 0.5 pkt. Zadania są ze sobą powiązane, ale można je rozwiązać niezależnie od siebie.

Trójkąt Pascala (0.5)

Jest to trójkątna tablica liczb, w której w n-tym rzędzie wartościami są współczynniki dwumianiu Newtona dla rozwinięcia $(a+b)^n$. Np w trzecim wierszu mamy 1 3 3 1 bo

$$(a+b)^3 = 1 * a^3 + 3 * a^2 * b + 3 * a * b^2 + 1 * b^3$$

. Taki trójkąt najłatwiej(?) wygenerować następującą procedurą:

1. pierwszy rząd składa się z 1
2. każdy następny rząd :
   - jest o 1 dłuższy
   - na końcach ma jedynki
   - liczby wewnątrz są sumą dwóch liczb stojących w wyższym wierszu bezpośrednio nad tą liczbą

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

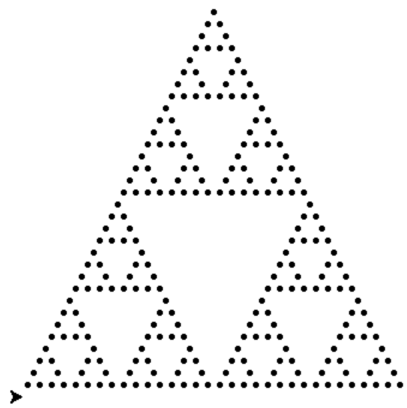
Zerknij na animację z wikipedii[1].

Zadanie polega na napisaniu procedury `pascal`, która dla zadanego parametru n generuje trójkąt o tym rozmiarze w postaci listy list(tupli). Wynikiem `pascal(n)` powinno być coś takiego `[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]`

[1] `https://pl.wikipedia.org/wiki/Trjkt_Pascala#/media/Plik:PascalTriangleAnimated2.gif`

Pascal vs Sierpiński (0.5)

Okazuje się, że trójkąt Sierpińskiego, znany z poprzedniej wprawki, w naturalny sposób pojawia się w trójkącie Pascala. Możemy to zaobserwować rysując elementy trójkątu Pascala jako czarne i białe kropki odpowiednio dla dla nieparzystych i parzystych współczynników. Np dla $n := 2^5 = 32$ wygląda to tak:



Zadanie polega na implementacji procedury przyjmującej jako parametr trójkąt Pascala w postaci z poprzedniego zadania i wykonującej żółwiem podobny rysunek. Jeśli nie masz zadania numer 1 to do testów możesz użyć listy ze skosowego pliku `pascal.triangle` Pomocne mogą się okazać funkcje

- `color()`
- `penup()`
- `dot()`
- `goto()`