- ## Opis problemu:

Matematyczna definicja:
Rozbicie zbioru (podział zbioru, partycja zbioru) - dla niepustego zbioru **A** to taka rodzina $\boldsymbol{\pi}$ niepustych podzbiorów tego zbioru, że każdy element zbioru **A** należy do dokładnie jednego podzbioru tej rodziny.

Liczba sposobów podziału skończonego zbioru n-elementowego wyraża się n-tą liczbą Bella $\boldsymbol{B_n}$.

Wzór Dobińskiego – w kombinatoryce wzór wyrażający liczbę podziałów zbioru n-elementowego.

$$\boldsymbol{B_n} = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}$$

Lista kilku pierwszych liczb Bella:

n	0	1	2	3	4	5	6	7	8	9	10	...
B_n	1	1	2	5	15	52	203	877	4140	21147	115975	...

Liczba Stirlinga dla podziałów $\left\{ \boldsymbol{n} \atop \boldsymbol{k} \right\}$ (często nazywana liczbą Stirlinga drugiego rodzaju) to liczba podziałów zbioru **n**-elementowego na dokładnie **k** bloki. Znów przyjmujemy, że $\left\{ 0 \atop 0 \right\} = 1$ oraz $\left\{ n \atop k \right\} = 0\ dla\ k < 0.$

Lista podziałów Z_4 na dwa bloki:

{0,1,2}{3}   {0,1}{2,3}
{0,1,3}{2}   {0,2}{1,3}
{0,2,3}{1}   {0,3}{1,2}
{1,2,3}{0}

- ## Opis algorytmu:

Generowanie **podziałów zbioru n-elementowego**:
  - Generuję tablicę n-elementową i wypełniam ją jedynkami,
  - wyszukuję pierwszą (zaczynając od prawej) pozycję na której mogę zwiększyć wartość o jeden, porównując ją do sąsiednich liczb (na lewo),
    (różnica między największa liczbą a liczba na obecnej pozycji nie może być większa niż jeden),
  - zwiększam wartość na znalezionej pozycji o jeden i otrzymuję jeden z podziałów,
  - wywołuję funkcję z argumentem (n), zaczynam wyszukiwanie od początku,

- w przypadku gdy nie można zwiększyć wartości na danej pozycji (różnica między największa liczbą na lewo a liczbą na obecnej pozycji wynosiła by więcej niż jeden), zamieniam liczbę na obecnej pozycji na jeden i wywołuję funkcję z argumentem (n-1) czyli przesuwam się na lewo,
- algorytm wywołuję się rekurencyjnie aż do mementu gdy zostanie wywołany z argumentem przekraczającym zakres pracy, oznacza to że wszystkie możliwe podziały zostały już utworzone.

- ## Fragment kodu:

Funkcja wywoływana rekurencyjnie:

```
void stirling(int i){
    if(i != 0){
        for(int k=0; k<=i-1; k++){
            if(tab[k]+1>c_max)c_max = tab[k]+1;
        }
        if(tab[i]<c_max){
            tab[i]++;
            show();
            stirling(n);
        }else{
            tab[i]=1;
            c_max=1;
            stirling(i-1);
        }
    }
};
```

Przykładowe wywołanie programu:

```
Generowanie wszystkich podzialow zbiodu n-elementowego
Podaj liczbe n:3
111
112
121
122
123
```

```
Generowanie wszystkich podzialow zbiodu n-elementowego
Podaj liczbe n:4
1111
1112
1121
1122
1123
1211
1212
1213
1221
1222
1223
1231
1232
1233
1234
```

- ## Wnioski:

  Program poprawnie generuje n-blokowe podziałów zbioru n-elementowego i wyświetla je w konsoli Windows. Dodatkowo waliduje dane wprowadzone przez użytkownika. Spełnia wszystkie założone kryteria.

Źródła:

[1]. Podziały — Mirek Rachelski smurf.mimuw.edu.pl/node/818

[2]. Rozbicie zbioru - pl.wikipedia.org/wiki/Rozbicie_zbioru