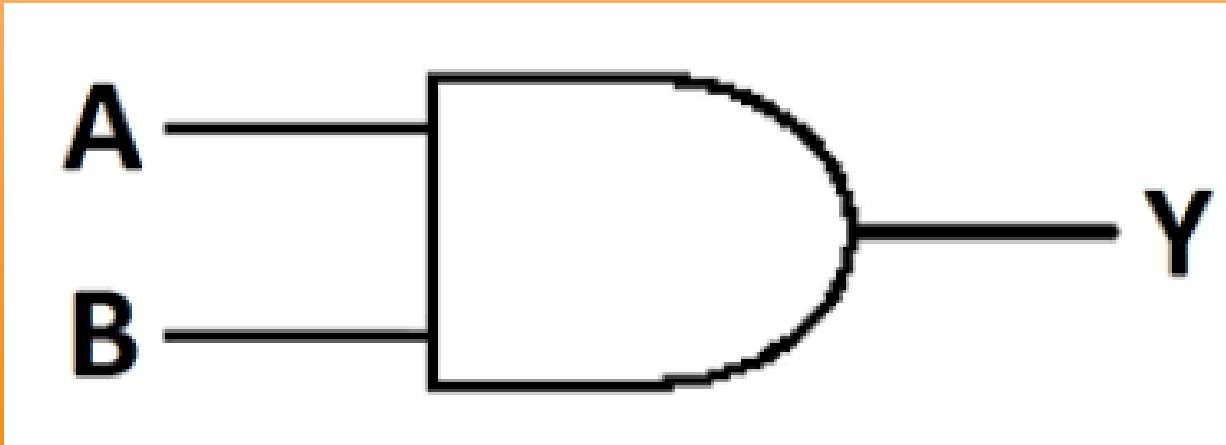


# Układy logiczne

i układy kombinacyjne

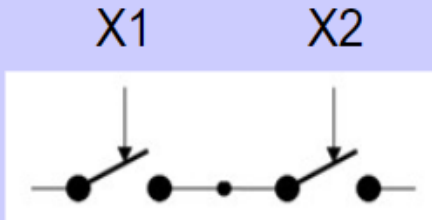
## Co to Układy logiczne ?

Układem **logicznym** nazywamy "czarną skrzynkę" posiadającą wejścia i wyjścia oraz nieznaną zawartość. W praktyce spotykamy zwykle **układy** binarne o  $n$  wejściach i wyjściach, przy czym sygnały na każdym wejściu i wyjściu mogą przyjmować tylko dwie wartości oznaczone zwykle symbolami 0 i 1.

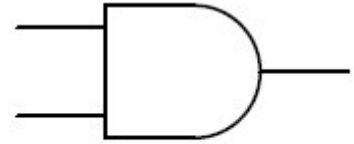


# Bramki logiczne:

**AND**- Przy działaniu bramki logicznej AND wynik 1 można otrzymać tylko w przypadku, kiedy oba wejścia będą równały się jedynce. Tego typu bramki mogą występować w wersjach trzywejściowych, czterowejściowych oraz o znacznie większej liczbie wejść. Należy pamiętać, że niezależnie od tego, ile wejść będzie znajdowało się w stanie wysokim – stan wysoki na wyjściu będzie możliwy tylko w przypadku, jeżeli na każdym wejściu będzie znajdowała się logiczna jedynka.



$$Y = X1 \bullet X2$$



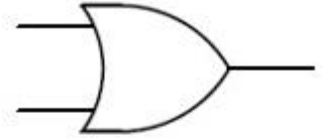
**AND**

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

# OR:

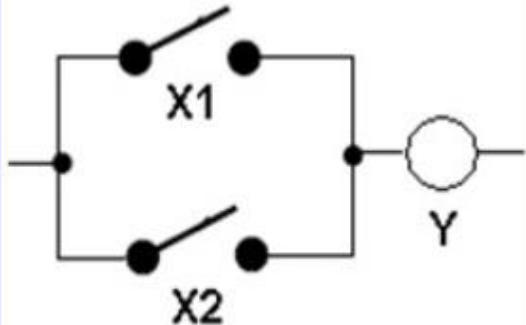
Na wyjściu bramki OR wynik o wartości 1 pojawi się zawsze w sytuacji, jeżeli chociaż jedno z wejść przyjmuje stan wysoki. Oznacza to również, że jeżeli pojawi się więcej niż jedno wejście o stanie wysokim – na wyjściu również pojawi się jedynka. Tak więc zero pojawi się na wyjściu wyłącznie w sytuacji, kiedy na wszystkich wejściach bramki również ustawione będzie zero.

Oczywiście wyjście jest zawsze tylko jedno. Typowy układ zawiera dwa wejścia, ale może posiadać ich nieskończenie wiele i będzie działał na takich samych zasadach



**OR**

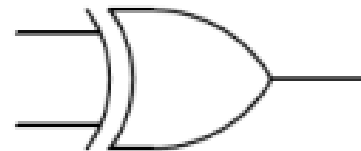
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



$$Y = X1 + X2$$

# XOR:

Bramka XOR to jedna z wyjątkowych funkcji, które nie należą już do grupy najczęściej stosowanych – podstawowych funkcji bramek logicznych. To bramka, która na wejściu ma zawsze dokładnie dwie zmienne. Uzyskuje wysoki stan zawsze, jeżeli tylko jeden ze stanów wejściowych jest równy logicznej jedynce. Stan niski na wyjściu pojawi się w sytuacji, jeżeli obie wartości na wejściu będą jednakowe. Bramka XOR wykonuje operacja dodawania bitów modulo-2.

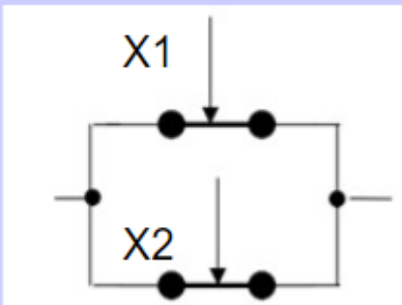


**XOR**

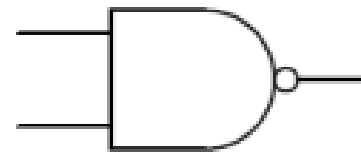
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

# NAND:

Działanie bramki NAND jest dokładnie odwrotne do działania bramki AND. Tak też można interpretować działanie bramki NAND – jest ono dokładnie takie, jakby na wejściu bramki AND zostały umieszczone odpowiednie wartości, na jej wyjściu pojawił się wynik, a następnie został on przepuszczony przez jeszcze bramkę NOT. Można również interpretować działanie NAND w taki sposób, jakby przed każdym z wejść na bramkę AND była bramka NOT. W uproszczeniu działanie bramki NAND można opisać tak – stan niski (0) pojawia się jedynie wtedy, jeżeli na wszystkich wejściach pojawi się stan wysoki (1). Warto zaznaczyć, że bramka NAND posiada nieograniczoną liczbę wejść.



$$Y = n(X1 \bullet X2)$$

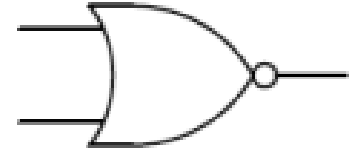


**NAND**

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

# NOR:

Przeoglądając poprzednie nazwy, można zauważyć pewną tendencję: bramki, których nazwy zostaną poprzedzone literą N, będą działały dokładnie odwrotnie w stosunku do swojego oryginału. Litera N (od angielskiego *not*, czyli "nie") oznacza bowiem negację (jak w przypadku AND i NAND). Działanie bramki NOR można określić jako całkowitą odwrotność działania bramki OR – można spojrzeć na nią, jak na działanie kolejno bramki OR na wszystkie sygnały wejściowe, jakie otrzymała, a następnie wprowadzenie jej sygnału wyjściowego na wejście bramki NOT – wynik będzie wskazywał na działanie bramki NOR. Oczywiście można również założyć, że przed każdym z wejść na bramkę OR zostanie zastosowana bramka NOT – wtedy każde wejście bramki OR będzie pochodziło z wyjścia bramki NOT. Upraszczając opis zasady działania, bramka NOR na wyjściu zawsze zwróci stan niski, z wyjątkiem sytuacji, w której wszystkie stany będą ustawione na wartość 0. Takie bramki mogą mieć nieskończoną ilość wejść.



**NOR**

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

X1 X2



$$Y = \overline{X1 + X2}$$

# XNOR:

bramka logiczna realizująca funkcję negacji alternatywy wykluczającej. Bramka ta neguje wynik bramki XOR, czyli zwraca fałsz, jeśli dokładnie jedno z wejść: A lub B jest prawdą, a w przeciwnym wypadku zwraca prawdę.

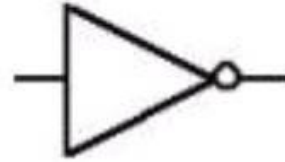


**XNOR**

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



# NOT:



TABLICA PRAWDY	
we	wyj
0	1
1	0

Jej działanie polega na negacji (odwróceniu) sygnału, który otrzyma na wejściu. W praktyce wygląda to następująco: jeżeli na wejściu zostanie nadany sygnał o wartości 1 (prawda), to na wyjściu pojawi się sygnał o wartości 0 (fałsz). Natomiast jeżeli na wejściu pojawi się sygnał 0, na wyjściu pojawi się 1.

## Schemat przekaźnikowy

stan „0”



stan „1”

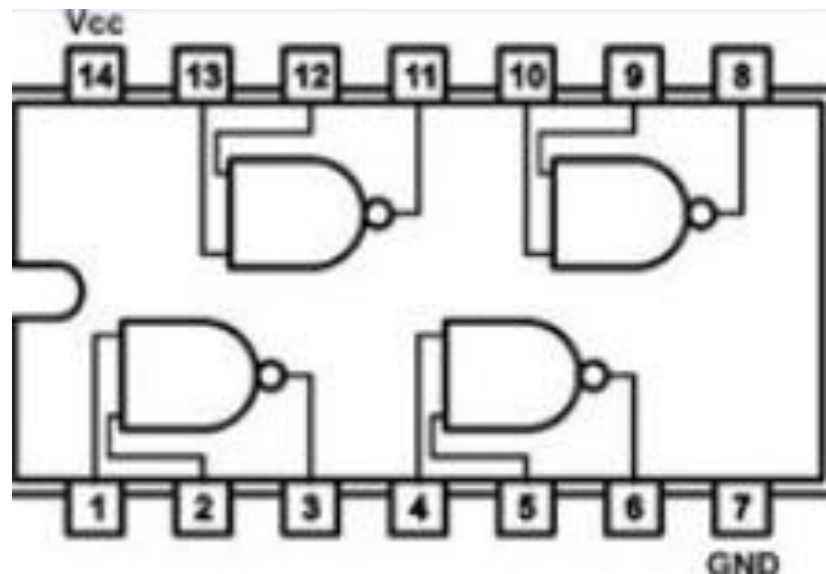
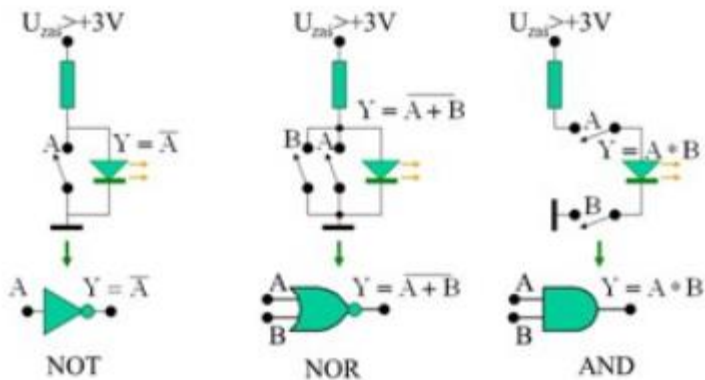


## Suma logiczna

$$Y = nX$$

# Zastosowanie w elektronice

## Modele prostych funkcyjów logicznych



# Układy logiczne



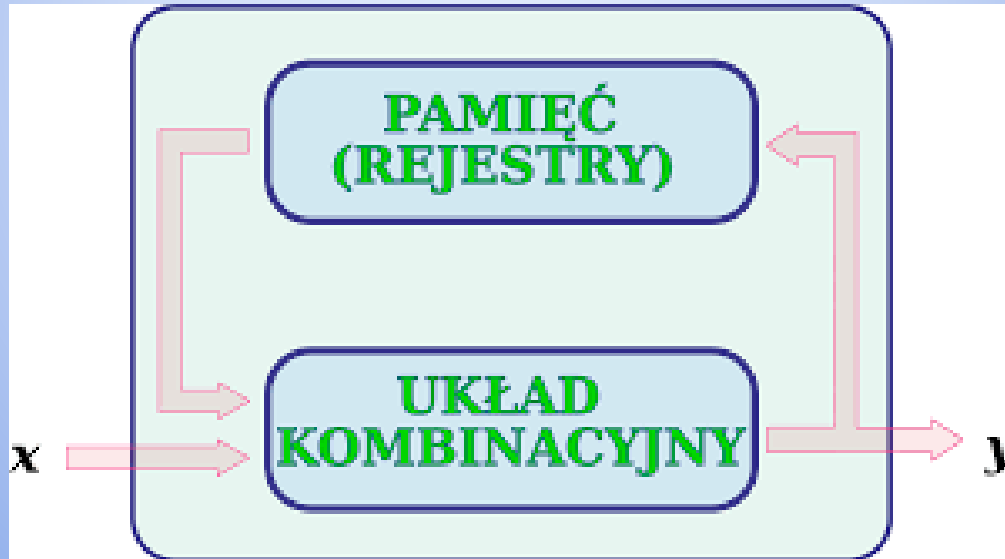
Kombinacyjne



Sekwencyjne

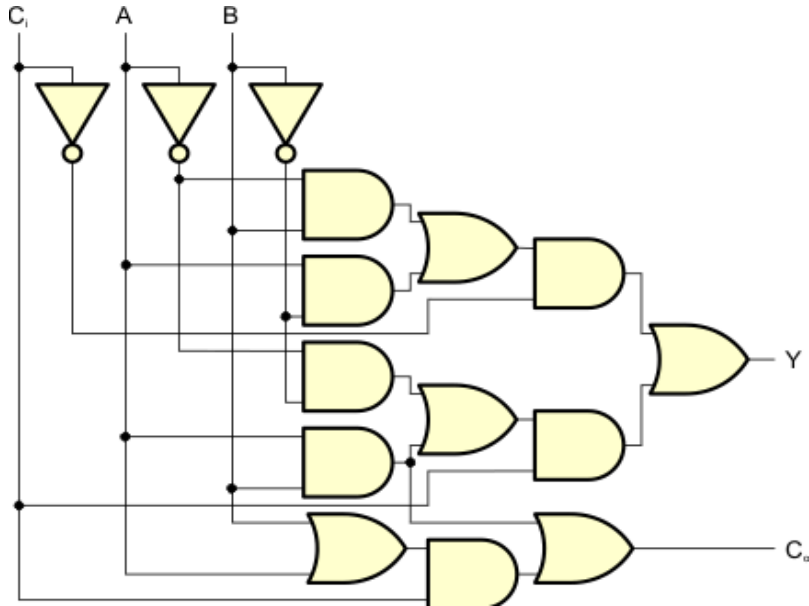
# Układy sekwencyjne

**Układ sekwencyjny** – rodzaj układów cyfrowych charakteryzujący się tym, że stan wyjść zależy od stanu wejść układu oraz od poprzedniego stanu, zwanego **stanem wewnętrznym**, pamiętanego w zespole rejestrów (pamięci).

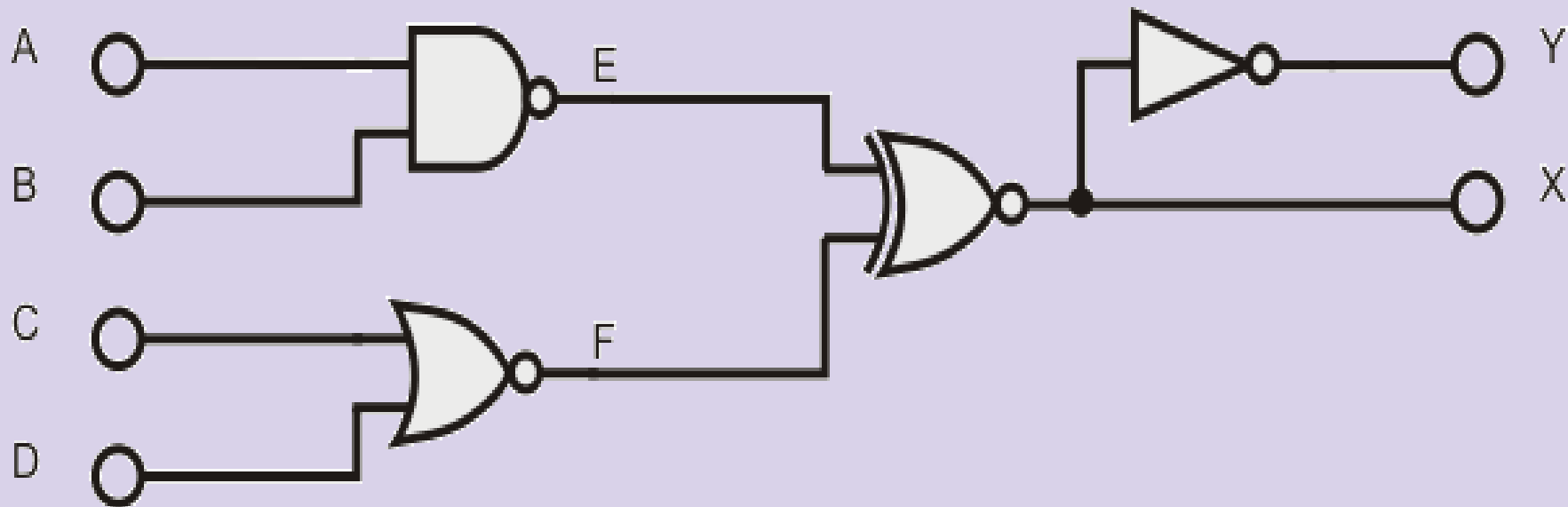


# Układy kombinacyjne

**Układ kombinacyjny** (ang. [combinational logic](#)) jest siecią logiczną zbudowaną z bramek cyfrowych, która realizuje pewną funkcję logiczną. Stan wyjść tej sieci zależy tylko od stanu jej wejść.



# Łączenie bramek logicznych



# Podział układów kombinacyjnych

## Układy kombinacyjne



### Komulatory



Multiplexer

Demultiplexer

### Konwertery kodów



Koder

Dekoder

Transkoder

### Bloki arytmetyczne



Sumator

Komparator

Alu

# Metody opisu układu kombinacyjnego

**Opis słowny** - jest pierwotną informacją o funkcjonowaniu układu, nie zawsze jest on jednoznaczny. Projektowanie układów kombinacyjnych na jego podstawie wymaga uściśleń.



# Metody opisu układu kombinacyjnego

**Tablica prawdy** - jest wykorzystywana przy opisie działania bramek logicznych. Wiersze tablicy zawierają wszystkie kombinacje sygnałów wejściowych układu oraz odpowiadające im stany wyjść układu. Jeśli ze względu na fizyczne działanie urządzenia pewne stany wejściowe nie mogą zaistnieć, w tablicy prawdy wpisuje się „-”, który traktuje się przy analizie i minimalizacji funkcji zależnie od potrzeb jako 1 lub 0. Mówimy, że w tym punkcie funkcja jest nieokreślona. Tablica prawdy to zwięzły i jednoznaczny opis funkcjonowania układu, bardzo przydatny przy jego projektowaniu.

# Tablica prawdy

## 1. Nagłówki

W tej sekcji należy od lewej strony rozpisywać nazwy/symbole wejść naszego układu.

## 2. Argumenty wejściowe

Tutaj należy rozpisać wszelkie możliwe kombinacje stanów logicznych na wejściach układu.

## 3. Wyniki

Ostatnią częścią tabeli są oczywiście wyniki. Tutaj zapisujemy stan logiczny, który pojawia się na wyjściu układu w przypadku, gdy na wejściach mamy stany podane obok.

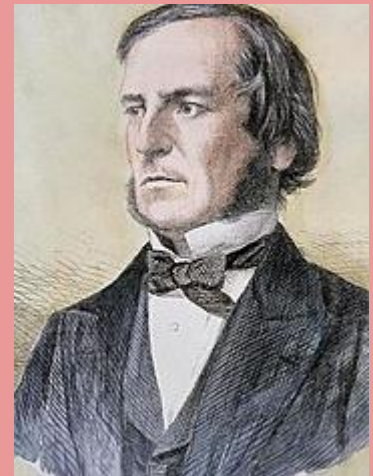
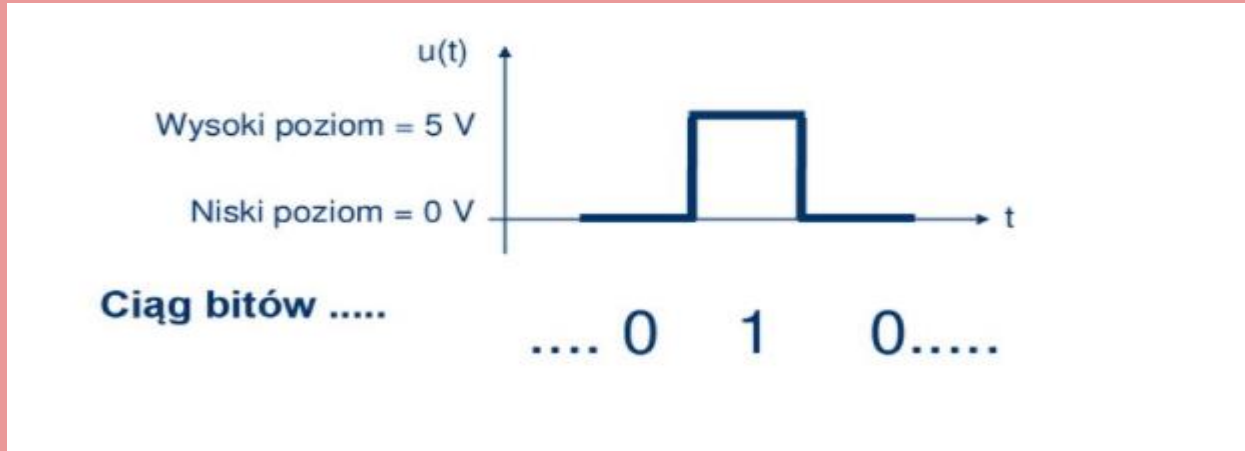
a	b	f
0	0	0
0	1	0
1	0	1
1	1	0

# Metody opisu układu kombinacyjnego

**Postać kanoniczna** - jest umownym sposobem opisu obiektów matematycznych. W postaci tej stosuje się pełne iloczyny czyli iloczyny wszystkich argumentów funkcji i pełne sumy czyli sumy wszystkich argumentów funkcji. Wyróżniamy kanoniczną postać sumy będącą sumą pełnych iloczynów funkcji dla których przyjmuje ona wartość 1 oraz kanoniczną postać sumy będącą sumą pełnych iloczynów dla których funkcja przyjmuje wartość 1 oraz kanoniczną postać iloczynu będącą iloczynem pełnych sum dla których funkcja przyjmuje wartość 0. Na podstawie postaci kanonicznej można budować układy kombinacyjne przy użyciu bramek, które jednak ze względu na rozbudowaną strukturę są drogie i zawodne.

# Dwuelementowa algebra Boole'a


**Algebra Boole'a** jest modelem matematycznym operacji na sygnałach binarnych reprezentujących sygnały elektryczne o dwóch wartościach : 0 i 1 . Wartości te są przyporządkowane dwóm poziomom napięcia wytwarzanego przez elektroniczne układy logiczne. Przyjmuje się że napięciu wysokiemu jest przyporządkowana wartość 1 a niskiemu wartość 0.





Dziękujemy za uwagę!!!


Prezentację robili :

🔊 prezentacja


 Dawid Otulakowski

 Kacper Kaczmarek

 Kuba Dębicki

 Oliwier Gireń

NIEDOSTĘPNY

 Antek Konarski

