QR code

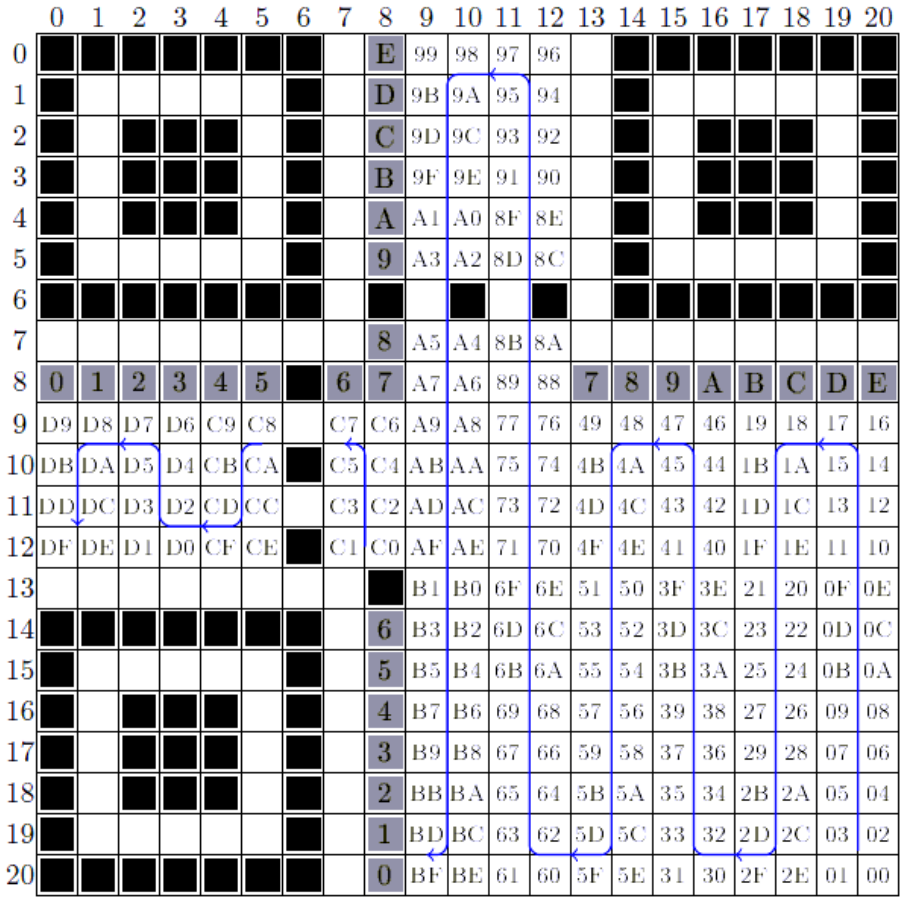
Liczba punktów:100
Limit czasu: 1s
Limit pamięci: 256MB

QR code jest standardem używanym powszechnie do przekazywania informacji. W dwuwymiarowym obrazku mogą być zakodowane różne dane. Standard określa 17 wielkości kodu oraz 4 poziomy korekcji błędów. Napisz obsługę najmniejszego kodu QR z najprostszym poziomem korekcji błędów.

Kod QR składa się z 3 części:

- niezmiennych wzorów pozycji i synchronizacji
- formatu danych (patrz informacja o maskowaniu)
- danych

W opisywanym kodzie obrazek ma rozmiar 21x21 komórek/bitów. Zapisuje się na nim 26*8 bitów, w tym 19*8 bitów ramki oraz 7*8 bitów korekcyjnych. Kolejność ciągu bitów jest opisana na rysunku poniżej liczbami od 0 do 207 (szesnastkowo). Na oznaczonych szarym kolorem polach jest zapisywany format danych, składa się on z 15 bitów.



Ramka składa się z następujących części: typ, liczba znaków, dane, terminator, dopełnienie do oktetu, dopełnienie do ramki.

Typ to 4 bity na których zapisany jest typ danych. Będziemy obsługiwali 3 typy:

- liczby 0001
- znaki alfanumeryczne (alnum) 0010
- bajty 0100

W polu *liczba znaków* zapisana jest liczba znaków jaka jest wpisana w dane. W zależności od typu składa się z:

- 10 bitów dla liczb
- 9 bitów dla alnum
- 8 bitów dla bajtów

Terminator, to 4 bity o wartości 0. Terminator może być mniejszy, jeśli długość dotychczasowego ciągu bitów jest większa od 19*8-4 (porównaj długość danych).
Jeśli długość dotychczasowego ciągu bitów nie jest wielokrotnością 8, to w pole *dopełnienie do oktetu* wpisujemy najmniejszą liczbę 0 tak, aby ciąg miał długość podzielną przez 8.
Jeśli dotychczasowy ciąg bitów nie ma długości 19*8, to w polu *dopełnienie do ramki* wpisujemy cyklicznie ciąg 11101100 00010001.
Kod korekcyjny jest wyznaczany przy użyciu dostarczonej biblioteki biblioteki.
Po wpisaniu ciągu 26*8 bitów do obrazka, część danych (czyli ramka i kod korekcyjny) jest maskowana (tj. zmieniają się kolory na przeciwne) przy użyciu jednej z 8 masek. Niech **l** to numer linii, a **k** to numer kolumny; jeśli podane wyrażenie jest prawdziwe, to piksel w podanym miejscu jest maskowany:

0. (**l**+**k**)%2==0 (format 111011111000100)
1. **l**%2==0 (format 111001011110011)
2. **k**%3==0 (format 111110110101010)
3. (**l**+**k**)%3==0 (format 111100010011101)
4. (**l**/2 + **k**/3)%2==0 (format 110011000101111)
5. ((**l***k**)%2 + (**l***k**)%3)==0 (format 110001100011000)
6. ((**l***k**)%2 + (**l***k**)%3)%2==0 (format 110110001000001)
7. ((**l**+**k**)%2 + (**l***k**)%3)%2==0 (format 110100101110110)

W zależności od typu dane zapisujemy następująco:

- **liczby** ciąg liczb grupujemy po 3 i każdą z grup zapisujemy przy użyciu 10 bitów zamieniając je w naturalny sposób. Jeśli ostatnia grupa zawiera 2 cyfry, to zapisujemy je na 7 bitach. Jeśli ostatnia grupa zawiera jedną cyfrę, to zapisujemy ją na 4 bitach.
- **alnum** ciąg znaków grupujemy po 2 i każdą z grup zapisujemy przy użyciu 11 bitów. Zamiana dokonywana jest na bazie przekształcenia zapisanej w tabelce poniżej (AB zamieniamy na 45*10+11 co daje ciąg 00111001101). Jeśli ostatnia grupa zawiera jeden znak, to zapisujemy go na 6 bitach.
- **bajty** ciąg bajtów kopiujemy w naturalny sposób do ciągu bitów

znak	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
kod	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

znak	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
kod	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
znak	U	V	W	X	Y	Z		$	%	*	+	-	.	/	:
kod	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44

Dane wejściowe

W pierwszej linii będą podane: typ danych (L - liczba, A - alnum, B - bajty), maska (cyfra od 0 do 7) i długość danych. W Drugiej linii będzie podana dana do zakodowania, dla liczb - ciąg cyfr, dla alnum - ciąg znaków, dla bajtów - ciąg bajtów zapisanych szesnastkowo, każdy bajt oddzielony spacją.

Dane wyjściowe

Przedstawić QRcode zorientowany tak, jak na obrazku w macierzy 21x21. Do zapisu znaków czarnych używamy **X**, do białych kropki.

Przykład

UWAGA Dla ułatwienia zrozumienia QRcode we wszystkich przykładach przedstawione są wersje z maskowaniem i bez maskowania. W rozwiązaniu należy wpisać jedynie wersję z maskowaniem.

Dane wejściowe:

```
B 0 10
0F FF 10 AE 00 84 49 ED 80 44
```

Dane wyjściowe:

```
XXXXXXX...X.X.XXXXXXX      XXXXXXX.......XXXXXXX
X.....X.....X.X.....X      X.....X..X.XX.X.....X
X.XXX.X.X.X...X.XXX.X      X.XXX.X.X...X.X.XXX.X
X.XXX.X.....X.X.XXX.X      X.XXX.X..X.XX.X.XXX.X
X.XXX.X..X.XX.X.XXX.X      X.XXX.X..XXX..X.XXX.X
X.....X..XXX..X.....X      X.....X...X...X.....X
XXXXXXX.X.X.X.XXXXXXX      XXXXXXX.X.X.X.XXXXXXX
........X.X.........       ........XXXX........
XXX.XXXXX.X.XXX...X..      XXX.XXXXX....XX...X..
XXXXX..XXX.X..XXX.X.X      X.X.XX..X....XX.XXXXX
..XX..X.X..X.XXX.X.X.      X..XX.X...XXXX.XXXXXX
XXX..X.XXX.XXXX..X..X      X.XX....X...X.XX...XX
XXX...XX..XX..X..XXX.      .X..X.XXX..XX...XX.XX
........XXX..X.X.X.X.      ........X.XX........
XXXXXXX.X.X.X.XXX.X.X      XXXXXXX.X......X.....
X.....X.X......X.XXXX      X.....X.XX.X.X....X.X
X.XXX.X.X...X.XXX....      X.XXX.X.X.X....X..X.X
X.XXX.X..XXX...X..XX.      X.XXX.X...X..X...XX..
X.XXX.X.X..X..X.....X      X.XXX.X.X.XXX...X.X..
X.....X.XX.XXX.X.X.X.      X.....X.X...X........
XXXXXXX.XXXX..X.X.XXX      XXXXXXX.XX.XX......X.
```

Dane wejściowe:
A 1 15
VI IGRZYSKA ETI

Dane wyjściowe:

```
XXXXXXX.X...X.XXXXXXX      XXXXXXX.XXXX..XXXXXXX
X.....X.X.X...X.....X      X.....X.X.X...X.....X
X.XXX.X.....X.X.XXX.X      X.XXX.X..XXX..X.XXX.X
X.XXX.X.......X.XXX.X      X.XXX.X.......X.XXX.X
X.XXX.X.XXXXX.X.XXX.X      X.XXX.X.X.....X.XXX.X
X.....X.XX.XX.X.....X      X.....X.XX.XX.X.....X
XXXXXXX.X.X.X.XXXXXXX      XXXXXXX.X.X.X.XXXXXXX
........X...X........      ........X...X........
XXX..XX.X.X..XXXX..XX      XXX..XX.XX.XXXXXX..XX
...XXX..X..X.XXXXXXX.      ...XXX..X..X.XXXXXXX.
..X...X...XXXXXXXXX.X      XX.XXXXXXX.........X.
...X.X...X.X.XX..XX..      ...X.X...X.X.XX..XX..
X..XXXXXXXXXXXXXXXX.       .XX...X.............X
........XXX...X..XXX.      ........XXX...X..XXX.
XXXXXXX..X...X.XXXX..      XXXXXXX...XXX.X....XX
X.....X.X...XX.X..XXX      X.....X.X...XX.X..XXX
X.XXX.X......XX.....X      X.XXX.X..XXXX..XXXXX.
X.XXX.X...X..XX...X..      X.XXX.X...X..XX...X..
X.XXX.X.X.XXX.X.XX.XX      X.XXX.X.XX...X.X..X..
X.....X.X.X.....XXX.X      X.....X.X.X.....XXX.X
XXXXXXX.X.XXXX.XX..XX      XXXXXXX.XX....X..XX..
```

Dane wejściowe:
L 2 41
398043948000487547615885458993768954341 08
Dane wyjściowe:

```
XXXXXXX....XX.XXXXXXX      XXXXXXX..X.X..XXXXXXX
X.....X.X.....X.....X      X.....X.XX..X.X.....X
X.XXX.X.......X.XXX.X      X.XXX.X..X..X.X.XXX.X
X.XXX.X.X..X..X.XXX.X      X.XXX.X.XX.XX.X.XXX.X
X.XXX.X..X....X.XXX.X      X.XXX.X.....X.X.XXX.X
X.....X.X..X..X.....X      X.....X.XX.XX.X.....X
XXXXXXX.X.X.X.XXXXXXX      XXXXXXX.X.X.X.XXXXXXX
.........X..X........      .....................
XXXXX.XXXX..XX.X.X.X.      XXXXX.XXX....X.X.X.X.
X..XX...X...XXXX..X.X      ....X...XX...XX.....X
....XXXXX.X.X.XXX.XXX      X..XXXXXXX...X.X..XX
X.XX...XX..X.X.X.....      ..X....XXX.XXX....X..
XX.X.XXX..XXX.XXX...X      .X...XXX.XXX..X.X.X.X
........XXX.XX..XX.X.      ........X.X..X.XXXXX.
XXXXXXX.X....XX..X.X.      XXXXXXX.XX..XXXX.XXX.
X.....X...XX.XXXX...X      X.....X..XXXXXX.X.X.X
X.XXX.X.XX.X.X..XX..X      X.XXX.X.X..XXX.XXXX.X
X.XXX.X.XXX..XX..XX..      X.XXX.X.X.X.XXXX.X...
X.XXX.X.XXX..X.X..X..      X.XXX.X.X.X.XX.......
X.....X.X...XX.X..XX.      X.....X.XX...X.....X.
XXXXXXX.X.XX.X.X..X..      XXXXXXX.XXXXXX.......
```

Powiązania

Liczba punktów:90
Limit czasu: 1-3 s
Limit pamięci: 256MB

W efekcie superhipertajnychohoh działań udało się zdobyć siatkę powiązań pomiędzy agentami obcych wywiadów. Dwóch agentów nie da się rozróżnić, jeśli zbiory powiązanych z nimi agentów są nierozróżnialne.

Głównodowodzący dał Tobie zadanie. Masz określić jak najwięcej cech agentów tak, aby stwierdzić którzy agenci są nierozróżnialni.

Dane wejściowe

W pierwszym wierszu znajdują się dwie liczby całkowite n i m oznaczające liczbę agentów (n<100) i liczbę powiązań. Agentów numerujemy od 0 do n-1. W kolejnych wierszach wpisane są opisy powiązań - pary agentów. Można założyć, że pary nie pojawiają się wielokrotnie.

Dane wyjściowe

W kolejnych liniach należy wpisać grupę nierozróżnialnych agentów. Każda linia musi kończyć się znakiem #. Agenci w linii muszą być posortowani rosnąco. Linie muszą być posortowane względem pierwszego agenta.

Przykład

Dane wejściowe:

```
6 7
0 1
2 0
5 4
5 3
4 2
0 5
3 4
```

Dane wyjściowe:

```
0 #
1 #
2 #
3 #
4 #
5 #
```

Agent 1 ma 1 powiązanie, agenci 2,3 mają 2 powiązania, agenci 0,4,5 mają 3 powiązania, więc grupami się różnią.
Agent 2 różni się od agenta 3, bo:

- różnią się odległości (wg powiązań) od agenta 1
- agent 3 jest w cyklu 3 elementowym z agentami 4,5, agent 2 nie jest w cyklu 3 elementowym

- agent 2 jest w cyklu 4 elementowym z agentami 0,4,5, agent 3 nie jest w cyklu 4 elementowym

Agent 0 różni się od 5, bo:

- jeśli rozróżniliśmy wcześniej agentów 2 i 3, to 0 sąsiaduje z agentem 2, a 5 nie
- agent 0 powiązany jest z agentem 1, a agent 5 nie

Dane wejściowe:
```
5 6
0 3
0 2
3 2
1 4
2 1
2 4
```

Dane wyjściowe:
```
0 1 3 4 #
2 #
```

Dane wejściowe:
```
5 5
0 1
1 2
2 3
3 4
4 0
```
Dane wyjściowe:
```
0 1 2 3 4 #
```

Dane wejściowe:
```
5 5
0 1
1 2
1 3
1 4
4 3
```
Dane wyjściowe:
```
0 2 #
1 #
3 4 #
```

Książka

Liczba punktów:80
Limit czasu: 1s
Limit pamięci: 256MB

Na planecie Krikkit mieszkał stary Hactar. Jego wrogowie, chociaż trochę honorowi, wygenerowali na Krikkit sieć, w której pozostawili największą księgę zawierającą odpowiedzi na najbardziej nurtujące pytania (oprócz oczywiście odpowiedzi na Wielkie pytanie o życie, wszechświat i całą resztę oraz liczbę Ramseya R(6,6)). W sieci (grafie na n<65 wierzchołkach) znajduje się wiele książek, których strony, co ciekawe, są trójkątne (książka to graf zawierający t trójkątów posiadających wspólną krawędź, tj. pełny graf trójdzielny $K_{1,1,t}$, gdzie t>0, zobacz dołączony rysunek). Komputer Hactar pragnie odnaleźć w sieci książkę z największą liczbą stron, ponieważ w przeciwnym przypadku zamieni się w chmurę pyłu. Pomóż Hactarowi!

Podpowiedź:
Przyjmijmy, że każdy graf zawiera książkę z 0 liczbą stron.
Testy:
-małe książki
-grafy pełne
-dokładnie książki
-większe książki
-szybkie

Wejście

W pierwszej linii liczba sieci (grafów) w teście, tj. liczba kolejnych linii. W kolejnych liniach liczba węzłów sieci (wierzchołków grafu).
Następnie po spacji macierz sąsiedztwa zapisana w jednej linii po spacjach (od lewej do prawej oraz z góry na dół).

Wyjście

W kolejnych liniach liczba stron książki, która ma największą liczbę stron i jest zawarta w sieci.

Przykład

Wejście
```
4
5 0 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0
4 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0
6 0 1 0 0 0 1 1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 0 0
5 0 1 0 0 1 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 0
```

Wyjście
```
2
2
3
0
```

Układanka

Liczba punktów:100
Limit czasu: 1s
Limit pamięci: 256MB

Napisz program układający z danego zestawu klocków oczekiwany kształt. Zarówno docelowy kształt jak i klocki składają się z kwadratów o boku jednostkowej długości.
Z każdego klocka można skorzystać co najwyżej raz, ale nie trzeba wykorzystywać wszystkich klocków. Klocki można obracać wokół własnej osi o wielokrotność 90 stopni oraz "odwracać na drugą stronę". Znaczy to, że klocek:

```
XXXX
X
```

można umieścić w docelowym kształcie w każdy z poniższych sposobów:

```
XXXX  XXXX   XX          XX
X          X  X    X  X    X
              X    X  X    X
X          X  X    X  X    X
XXXX  XXXX      XX  XX
```

Wejście

W pierwszej linii podane będą:
w – szerokość planszy,
h – wysokość planszy,
n – liczba klocków.
Następnie pojawi się n opisów klocków, każdy postaci:
w_i h_i
$k_{00}k_{10}$...
$k_{01}k_{11}$...
...
gdzie w_i to szerokość klocka, h_i to jego wysokość. Kolejne h_i linii po w_i znaków opisują wygląd klocka – jeżeli k_{xy} jest kropką oznacza to pusty kwadrat zaś X oznacza kwadrat należący do klocka.
Na koniec pojawi się h linii po w znaków, opisujących kształt do utworzenia. Kropka oznacza, że dane pole powinno pozostać puste, X oznacza że pole powinno zostać zasłonięte klockiem.
Liczba klocków nie przekroczy 15 a ich wymiary nie będą większe niż 5x5.
Wymiary planszy nie przekroczą 24x24.

Wyjście

Na wyjściu należy wypisać docelowy kształt, zastępując znaki X literami odpowiadającymi klockom, które zajmują dane pole. Znak A odpowiada pierwszemu klockowi na wejściu, B drugiemu itp.

Przykład

Wejście

```
8 8 12
3 3
.XX
XX.
.X.
5 1
XXXXX
2 4
X.
X.
X.
XX
4 2
XX..
.XXX
2 3
XX
XX
X.
3 3
XXX
.X.
.X.
3 2
X.X
XXX
3 3
X..
X..
XXX
3 3
X..
XX.
.XX
3 3
.X.
XXX
.X.
4 2
..X.
XXXX
3 3
XX.
.X.
.XX
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXX..XXX
XXX..XXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
```

Wyjście

```
ACCCCLLK
AAAICLKK
HAIILLEK
HII..EEK
HHH..EEF
GGJDDFFF
GJJJDDDF
GGJBBBBB
```

Wejście

```
10 6 12
3 3
.XX
XX.
.X.
5 1
XXXXX
2 4
X.
X.
X.
XX
4 2
XX..
.XXX
2 3
XX
XX
X.
3 3
XXX
.X.
.X.
3 2
X.X
XXX
3 3
X..
X..
XXX
3 3
X..
XX.
.XX
3 3
.X.
XXX
.X.
4 2
..X.
XXXX
3 3
XX.
.X.
.XX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

Wyjście

```
ABBBBBLHHH
AAADDDLLLH
CADDKKKKLH
CEEEIFKJGG
CEEIIFJJJG
CCIIFFFJGG
```

Dopasowanie wzorca

Liczba punktów:40
Limit czasu: 1s
Limit pamięci: 256MB

Napisz program sprawdzający czy napis zawiera podany wzorzec.
Napis zawiera wzorzec, jeżeli możliwe jest usunięcie z napisu zera lub więcej znaków z jego początku oraz zera lub więcej znaków z jego końca tak, aby pozostałe znaki były takie same jak wzorzec. Dodatkowo, znak . (kropka) pasuje do dowolnej jednej litery napisu zaś znak * (gwiazdka) pasuje do dowolnej liczby (w tym zera) liter napisu. I tak:
abcdef zawiera wzorzec abc
abcdef zawiera wzorzec def
abcdef zawiera wzorzec bcd
abcdef nie zawiera wzorca acd
abcdef zawiera wzorzec a.c
abcdef zawiera wzorzec a*c
abcdef zawiera wzorzec a*b
abcdef zawiera wzorzec a*f
abcdef zawiera wzorzec b.*d
abcdef nie zawiera wzorca b.*c
abcdef zawiera wzorzec a*.*f
abcdef zawiera wzorzec a*b*c*f
Wielkość liter ma znaczenie podczas porównywania (małe a jest innym znakiem niż wielkie A).

Wejście

W pierwszej linii podana będzie liczba testów. Następnie podane będą, w osobnych liniach, testy. Każdy test będzie składał się z wzorca i oddzielonego od niego spacją napisu. Liczba testów nie przekroczy 1000. Długości napisów nie przekroczą 100000 zaś długości wzorców będą nie większe niż 10000. Napis będzie składał się wyłącznie z cyfr oraz małych i wielkich liter. Wzorce dodatkowo mogą zawierać znaki . (kropka) oraz * (gwiazdka).

Wyjście

Dla każdego testu należy wypisać, w osobnej linii, słowo **TAK** jeżeli napis zawiera wzorzec lub słowo **NIE** jeżeli napis nie zawiera wzorca.

Przykład

Wejście	**Wyjście**
29	
abcdef abc	TAK
abcdef def	TAK
abcdef bcd	TAK
abcdef acd	NIE
abcdef a.c	TAK
abcdef a*c	TAK
abcdef a*b	TAK
abcdef a*f	TAK
abcdef b.*d	TAK
abcdef b.*c	NIE
abcdef a*.*f	TAK
abcdef a*b*c*f	TAK
aaaaaaaaaa a*a*a*a*a*a*a*a*a*a	TAK
aaaaaaaaaa	TAK
aaaaaaaaaa	NIE
aaaaaaaaaa .....*********	TAK
aaaaaaaaaa .....********.	TAK
XxXxXxXxXxX XxXxXxXxX	TAK
XxXxXxXxXxX .XxXxXxXxX	TAK
XxXxXxXxXxX XxXxXxXxX.	TAK
XxXxXxXxXxX .XxXxXxXxX.	NIE
XxXxXxXxXxX *XxXxXxXxX	TAK
XxXxXxXxXxX XxXxXxXxX*	TAK
XxXxXxXxXxX *XxXxXxXxX*	TAK
XxXxXxXxXxX *	TAK
XxXxXxXxXxX .	TAK
XxXxXxXxXxX x	TAK
XxXxXxXxXxX X	TAK
XxXxXxXxXxX y	NIE