

Ćwiczenia 5

Wskaźniki, Tablice dwuwymiarowe.

„dane” ↔ „wskaźniki”

```
char c = 'A';
```

```
char *wc = &c;
```

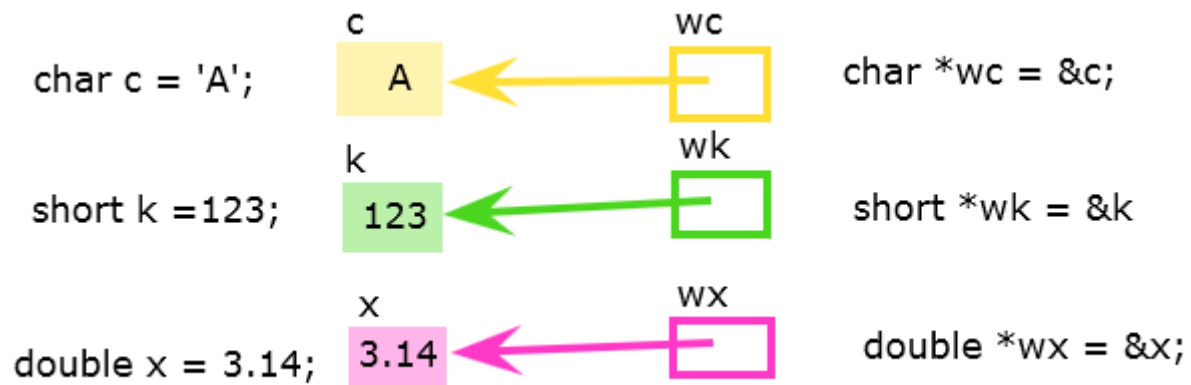
```
short k = 123;
```

```
short *wk = &k
```

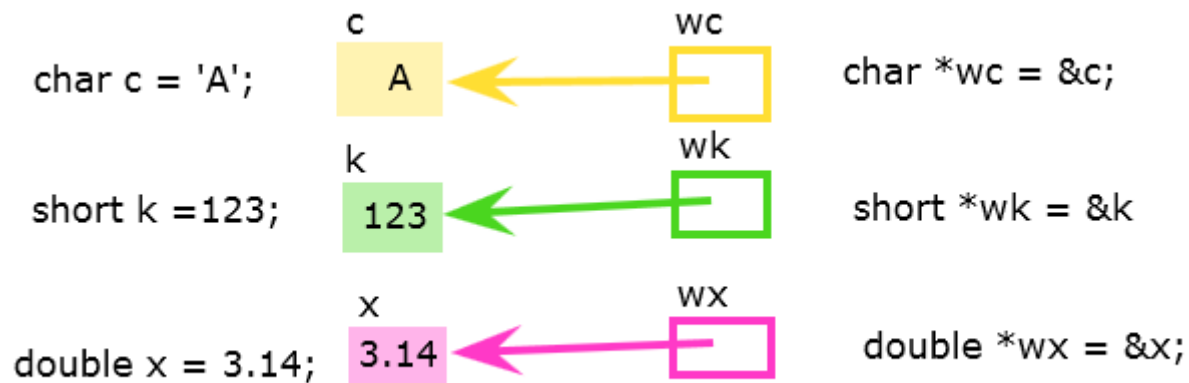
```
double x = 3.14;
```

```
double *wx = &x;
```

„dane” ↔ „wskaźniki”



„dane” ↔ „wskaźniki”



- zmienna typu char (1 bajt)
- zmienna typu short (2 bajty)
- zmienna typu double (8 bajtów)

- wskaźniki na char, short, double (zawsze 4 bajty)
-
-

„wskaźnik” \leftrightarrow „tablica”

```
char tc[3] = { 'A', 'B', 'C' };
```

3 – elementowa tablica znaków

```
char *w = tc;
```

wskaźnik na pierwszy element tablicy znaków

```
char (*wt)[3] = &tc;
```

wskaźnik na całą 3 –elementową tablicę znaków

„wskaźnik” \leftrightarrow „tablica”

```
char tc[3] = { 'A', 'B', 'C' };
```

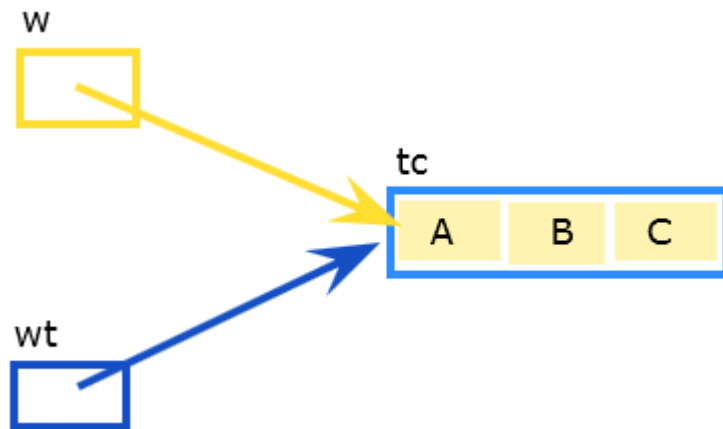
3 – elementowa tablica znaków

```
char *w = tc;
```

wskaźnik na pierwszy element tablicy znaków

```
char (*wt)[3] = &tc;
```

wskaźnik na całą 3 –elementową tablicę znaków



„wskaźnik” \leftrightarrow „tablica”

```
char tc[3] = { 'A', 'B', 'C' };
```

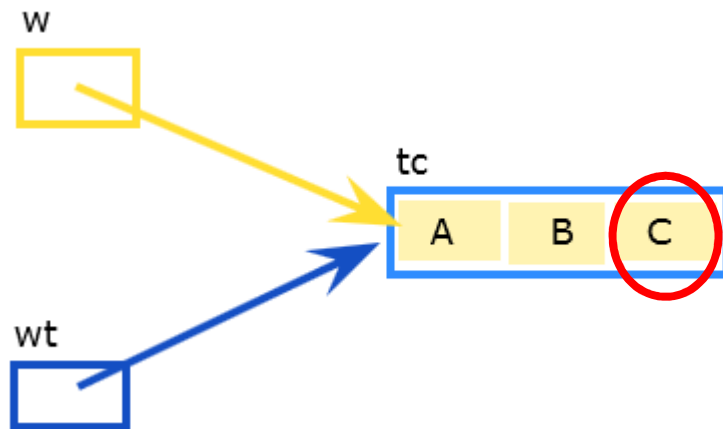
3 – elementowa tablica znaków

```
char *w = tc;
```

wskaźnik na pierwszy element tablicy znaków

```
char (*wt)[3] = &tc;
```

wskaźnik na całą 3 –elementową tablicę znaków



„wskaźnik” \leftrightarrow „tablica”

```
char tc[3] = { 'A', 'B', 'C' };
```

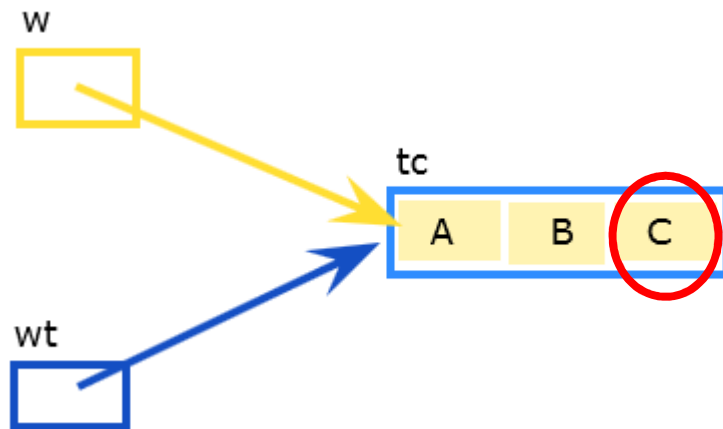
3 – elementowa tablica znaków

```
char *w = tc;
```

wskaźnik na pierwszy element tablicy znaków

```
char (*wt)[3] = &tc;
```

wskaźnik na całą 3 –elementową tablicę znaków



`tc[2]`

`*(w+2)`

`__(*wt+2)`

„wskaźnik” \leftrightarrow „tablica”

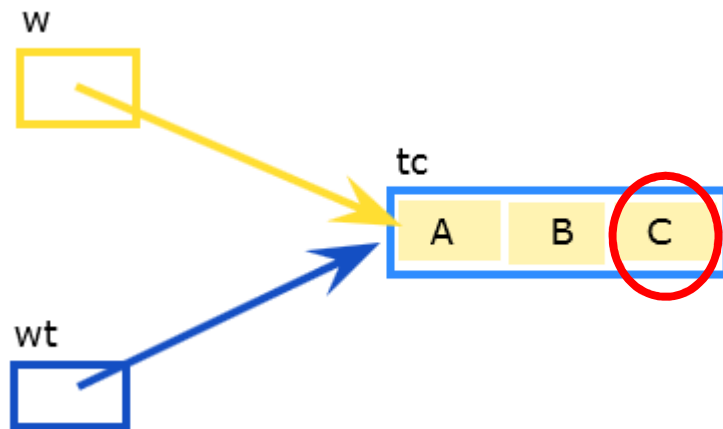
```
char tc[3] = { 'A', 'B', 'C' };
```

```
char *w = tc;
```

```
char (*wt)[3] = &tc;
```

W języku C/C++ nazwa tablicy jest stałą typu „wskaźnik na pierwszy element tablicy”

W języku C/C++ operator wyłuskania `*` można stosować zamiennie z operatorem indeksowania `[]`



`tc[2]`

`*(w+2)`

`__(*wt+2)`

„wskaźnik” \leftrightarrow „tablica”

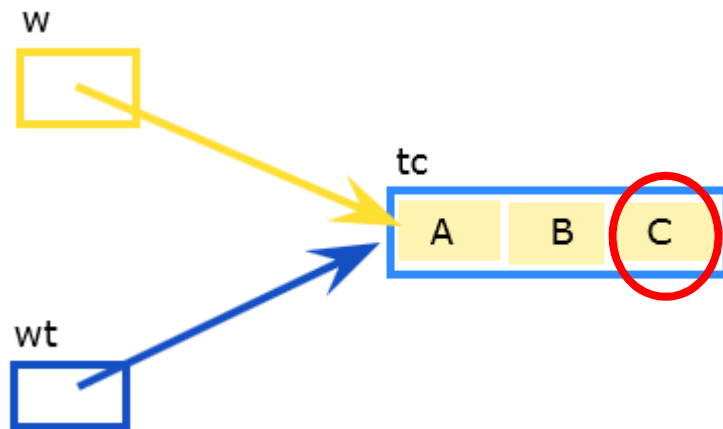
```
char tc[3] = { 'A', 'B', 'C' };
```

```
char *w = tc;
```

```
char (*wt)[3] = &tc;
```

W języku C/C++ nazwa tablicy jest stałą typu „wskaźnik na pierwszy element tablicy”

W języku C/C++ operator wyłuskania `*` można stosować zamiennie z operatorem indeksowania `[]`



`tc[2]`

`*(tc+2)`

`*(w+2)`

`w[2]`

`__(*wt+2)`

`wt[0][2]`

„wskaźnik” \leftrightarrow „tablica”

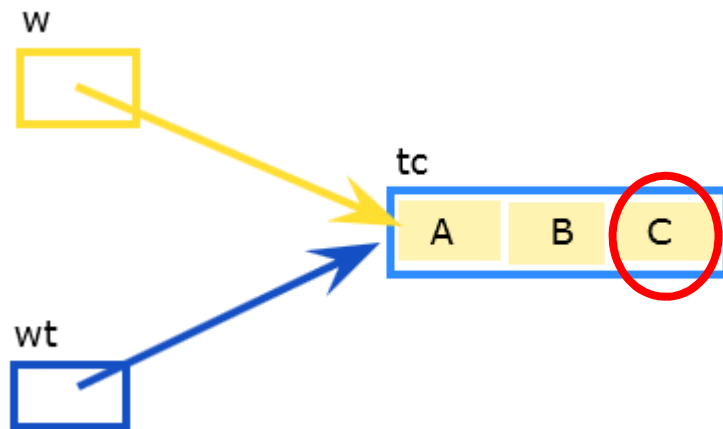
```
char tc[3] = { 'A', 'B', 'C' };
```

```
char *w = tc;
```

```
char (*wt)[3] = &tc;
```

W języku C/C++ nazwa tablicy jest stałą typu „wskaźnik na pierwszy element tablicy”

W języku C/C++ operator wyłuskania `*` można stosować zamiennie z operatorem indeksowania `[]`



`tc[2]`

`*(tc+2)`

`*(w+2)`

`w[2]`

`*(wt+2)`

`wt[0][2]`

`(*wt)[2]`

`*(wt[0]+2)`

```
int (*wskT)[3];
```

```
int *tabW[3];
```

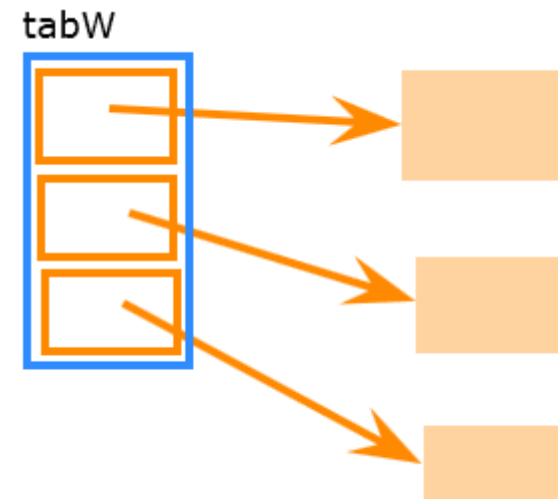
```
int (*wskT)[3];
```

wskaźnik na 3 elementową tablicę
liczb typu int



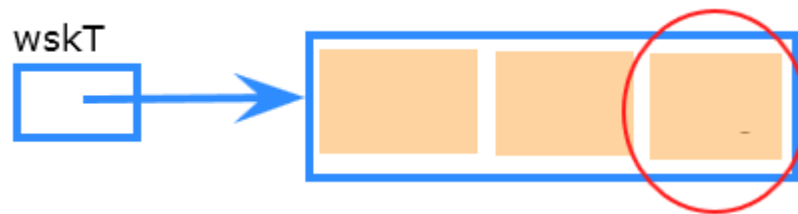
```
int *tabW[3];
```

3 elementowa tablica wskaźników na
liczby typu int



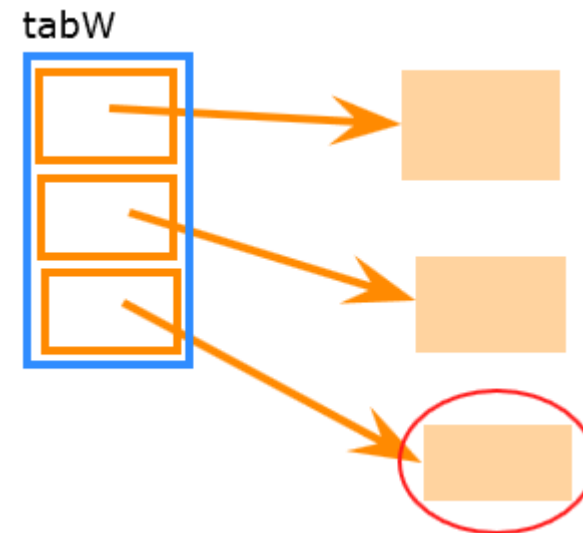
```
int (*wskT)[3];
```

wskaźnik na 3 elementową tablicę
liczb typu int



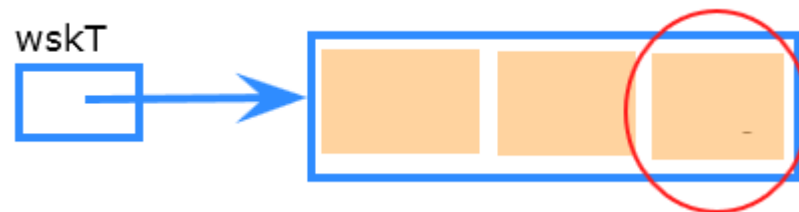
```
int *tabW[3];
```

3 elementowa tablica wskaźników na
liczby typu int



`int (*wskT)[3];`

wskaźnik na 3 elementową tablicę
liczb typu int



`(*wskT)[2]`

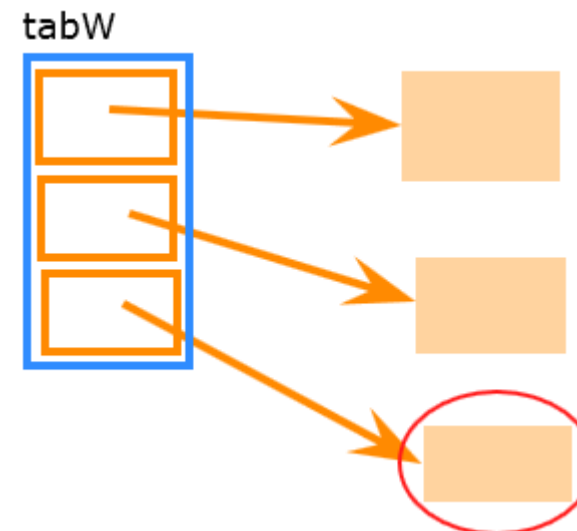
`*(wskT + 2)` `= *(*wskT + 2)`

`wskT[0][2]`

`*(wskT[0] + 2)`

`int *tabW[3];`

3 elementowa tablica wskaźników na
liczby typu int



`*(tabW[2])`

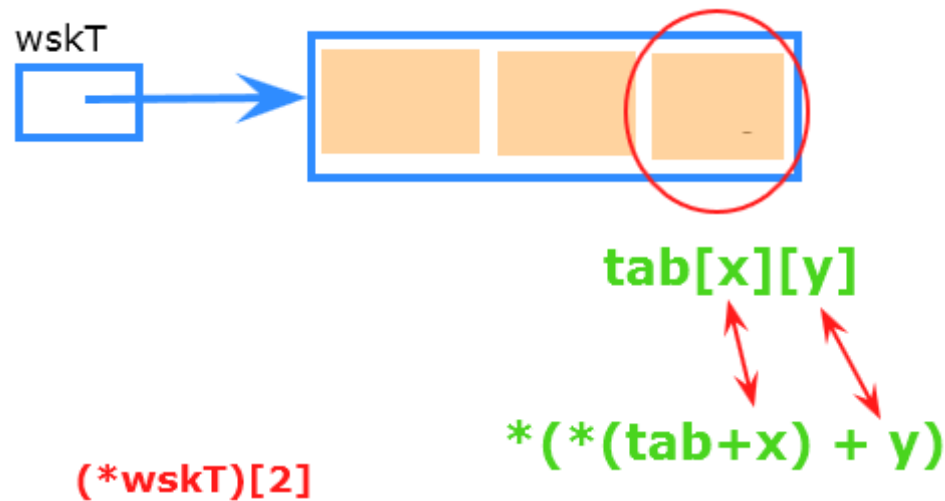
`**tabW + 2` `= *(*tabW + 2) + 0`

`tabW[2][0]`

`*(tabW[2])[0]`

`int (*wskT)[3];`

wskaźnik na 3 elementową tablicę
liczb typu int



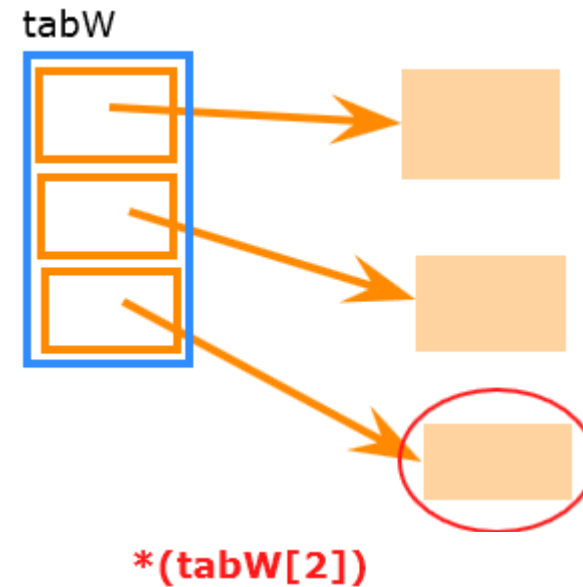
`*(*wskT + 2) = *(* (wskT+0) + 2)`

`wskT[0][2]`

`* (wskT[0]+2)`

`int *tabW[3];`

3 elementowa tablica wskaźników na
liczby typu int



`** (tabW+2) = *(* (tabW+2) + 0)`

`tabW[2][0]`

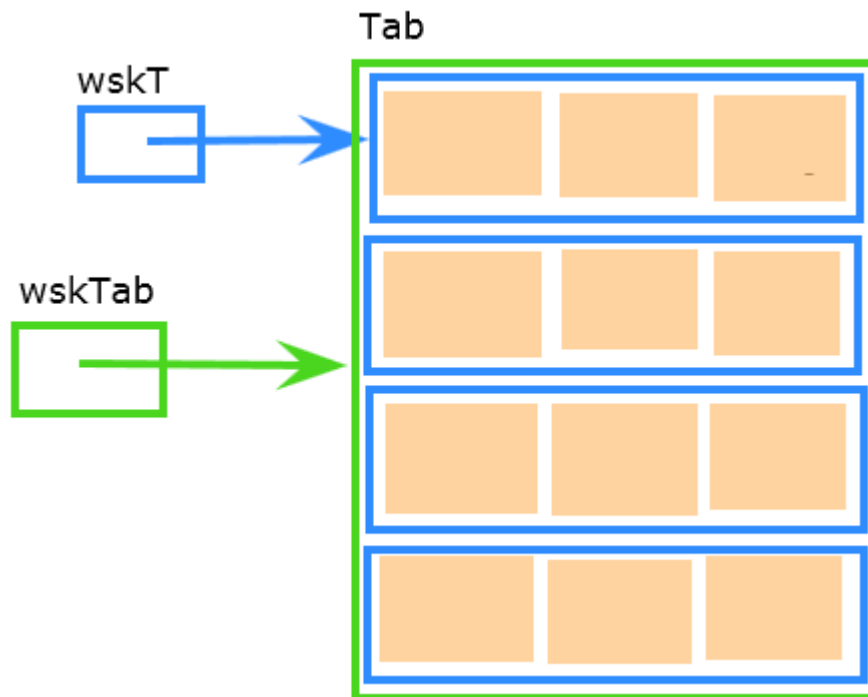
`(* (tabW+2))[0]`


```
int Tab[4][3];
```

„tablica dwuwymiarowa”

```
int Tab[4][3];
```

Tab - 4-elementowa tablica, której elementami są 3-elementowe tablice liczb typu int



„tablica dwuwymiarowa”

```
int Tab[4][3];
```

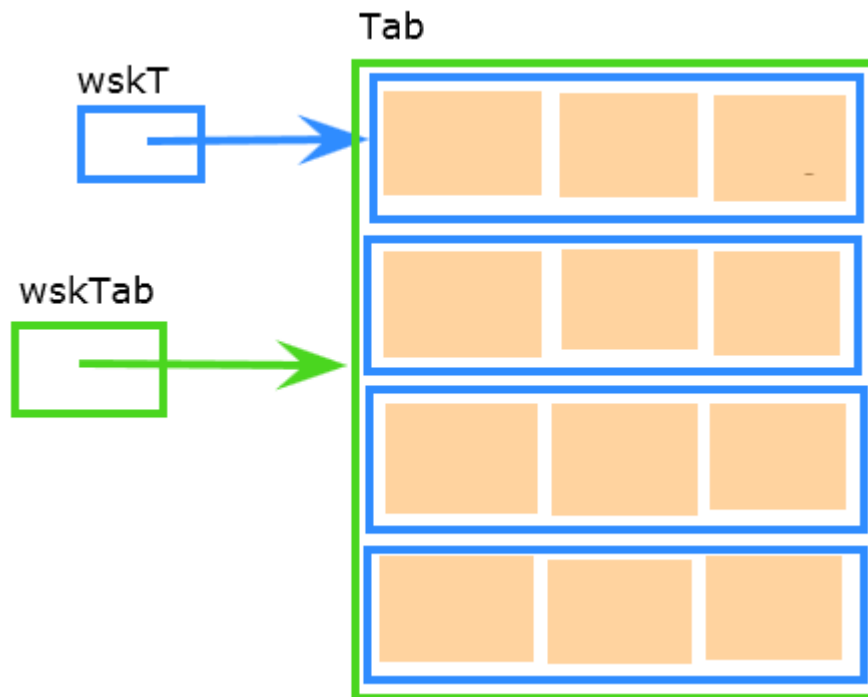
```
int (*wskT)[3];
```

```
int (*wskTab)[4][3];
```

Tab - 4-elementowa tablica, której elementami są 3-elementowe tablice liczb typu int

wskT - wskaźnik na pierwszy element tablicy, której elementami są 3-elementowe tablice liczb typu int

wskTab - wskaźnik na całą dwuwymiarową tablicę 4 wierszów i 3 kolumnach

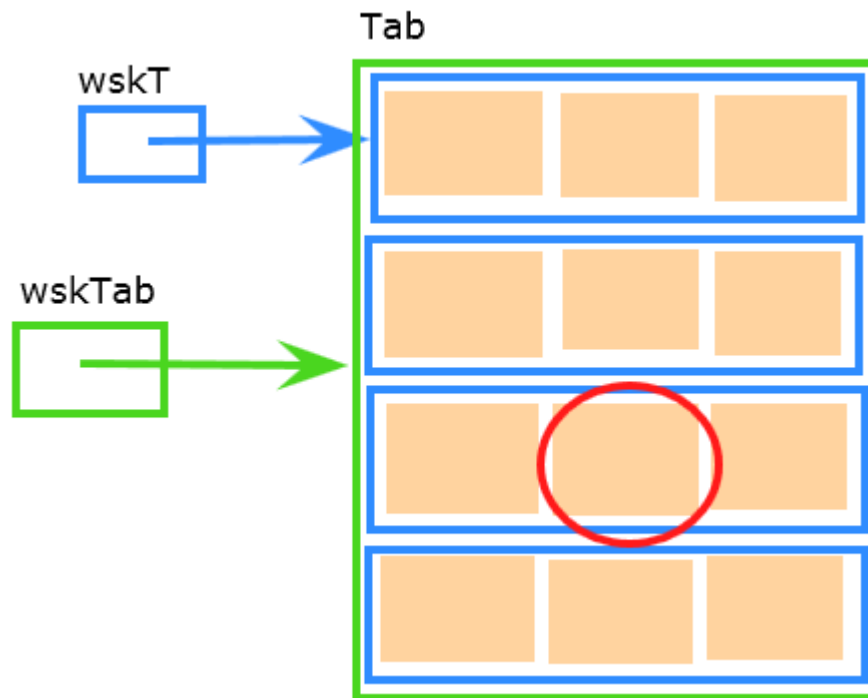


„tablica dwuwymiarowa”

```
int Tab[4][3];
```

```
int (*wskT)[3];
```

```
int (*wskTab)[4][3];
```



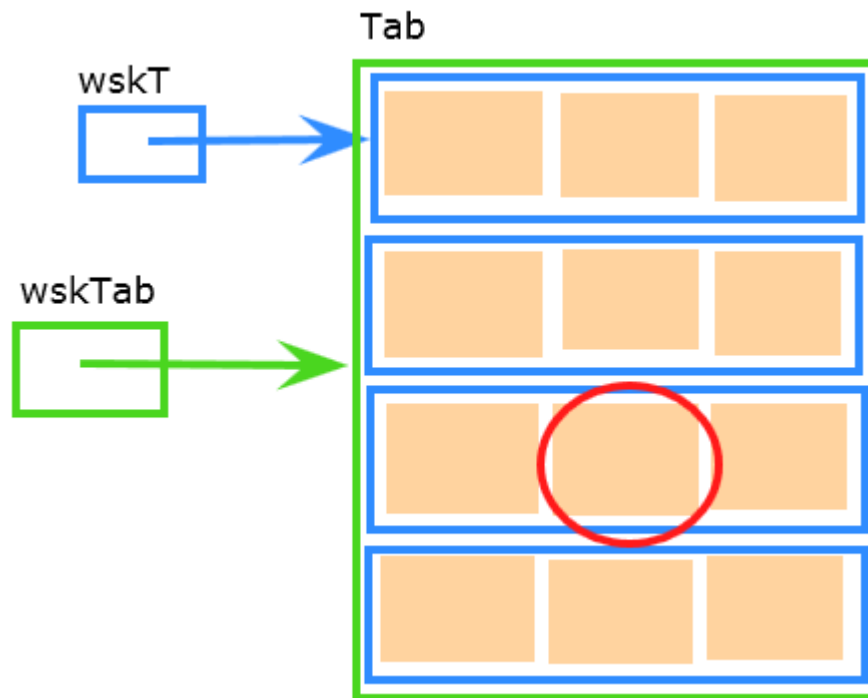
„tablica dwuwymiarowa”

```
int Tab[4][3];
```

`Tab[2][1]`

```
int (*wskT)[3];
```

```
int (*wskTab)[4][3];
```



`*(*wskT+2)+1`

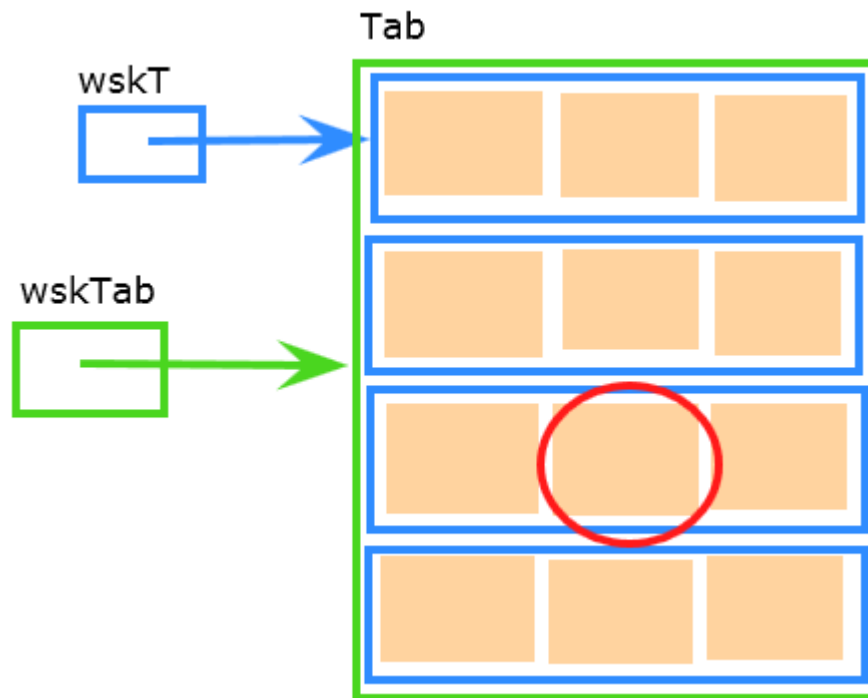
`(*wskTab)[2][1]`

„tablica dwuwymiarowa”

```
int Tab[4][3];
```

```
int (*wskT)[3];
```

```
int (*wskTab)[4][3];
```



`Tab[2][1]`

`*(*(Tab+2)+1)`

`*(Tab[2]+1)`

`*(Tab+2)[1]`

`*(*(wskT+2)+1)`

`wskT[2][1]`

`*(wskT[2]+1)`

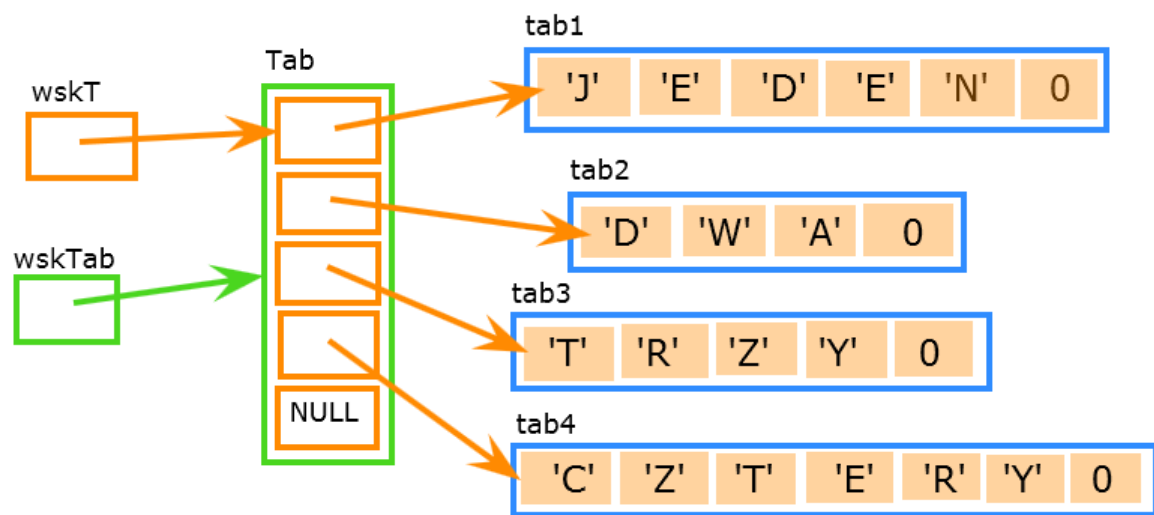
`*(wskT+2)[1]`

`(*wskTab)[2][1]`

`wskTab[0][2][1]`

.....

`*(**wskTab + 2)+1)`

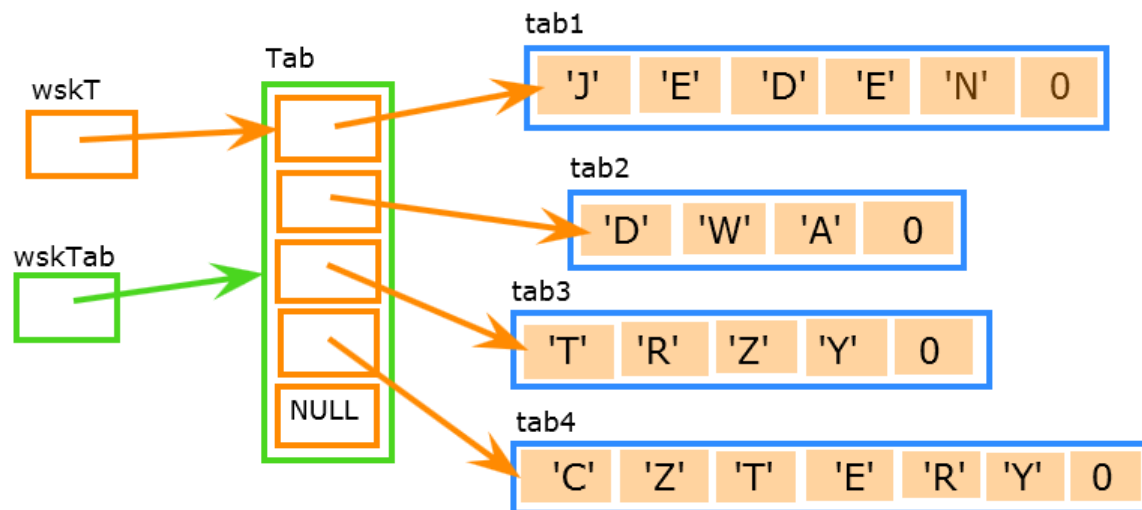


```
char tab1[] = "JEDEN";  
char tab2[] = "DWA";  
char tab3[] = "TRZY";  
char tab4[] = "CZTERY";
```

```
char *Tab[5] = {tab1, tab2, tab3, tqb4, NULL};  
char *(*wskTab)[5] = &Tab;
```

char **wskT;

wskT = Tab;

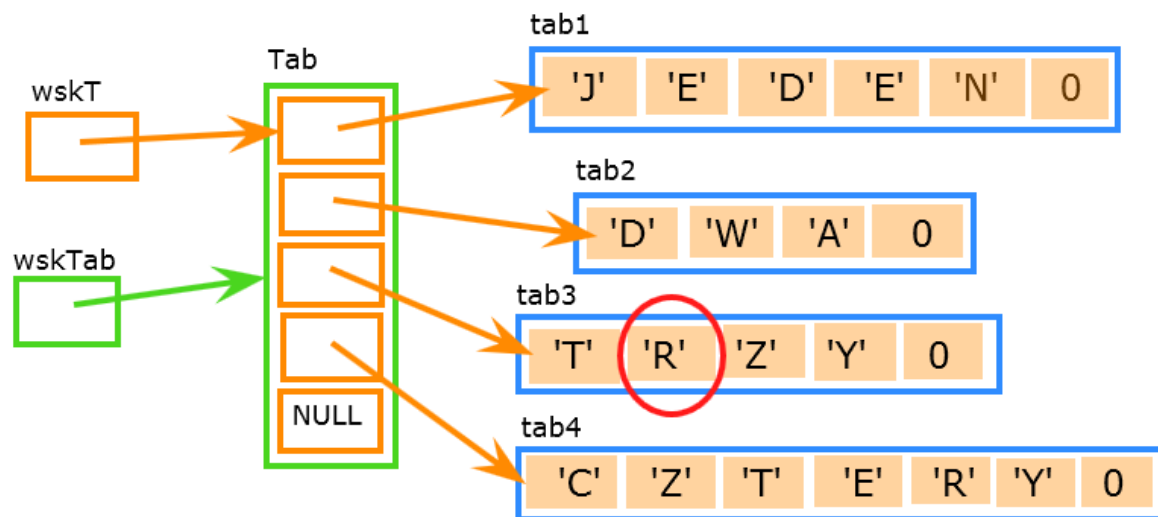



```
char tab1[] = "JEDEN";  
char tab2[] = "DWA";  
char tab3[] = "TRZY";  
char tab4[] = "CZTERY";
```

```
char *Tab[5] = {tab1, tab2, tab3, tqb4, NULL};  
char *(*wskTab)[5] = &Tab;
```

char **wskT;

wskT = Tab;



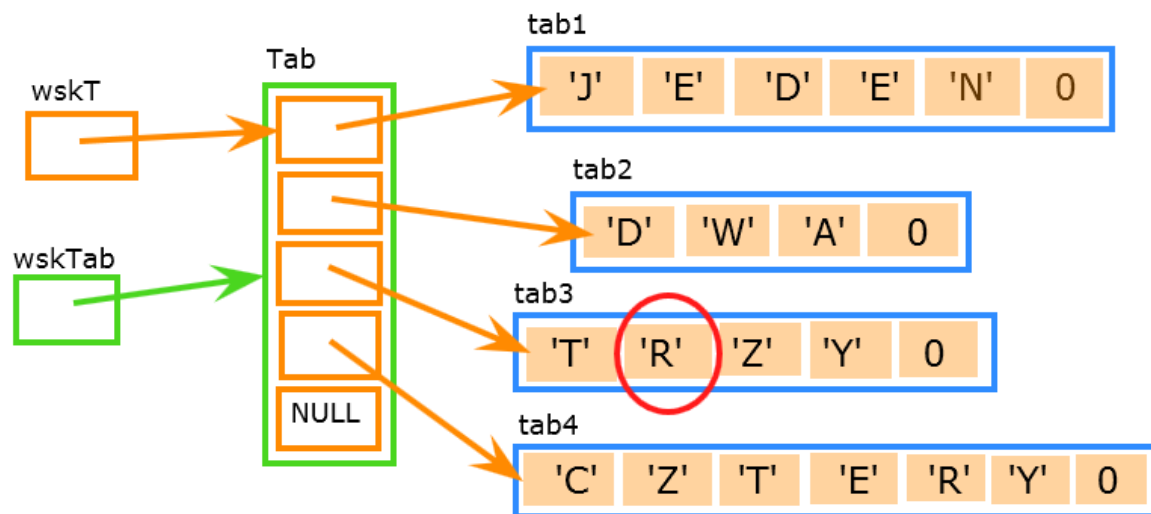
tablica dwuwymiarowa?

```
char tab1[] = "JEDEN";  
char tab2[] = "DWA";  
char tab3[] = "TRZY";  
char tab4[] = "CZTERY";
```

```
char *Tab[5] = {tab1, tab2, tab3, tab4, NULL};  
char *(*wskTab)[5] = &Tab;
```

```
char **wskT;
```

```
wskT = Tab;
```



Tab[2][1]

`*(*(Tab+2)+1)`
`*(Tab[2]+1)`
`(*(Tab+2))[1]`

***(*(wskT+2)+1)**

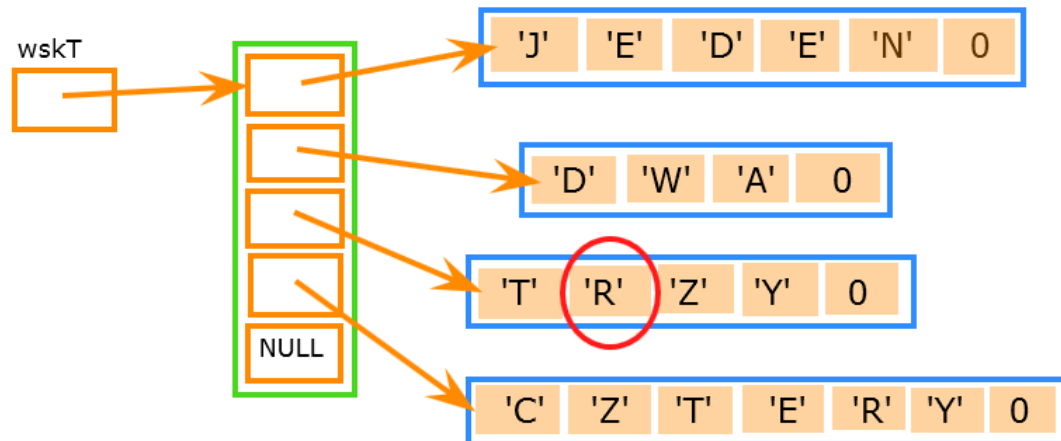
`wskT[2][1]`
`*(wskT[2]+1)`
`(*(wskT+2))[1]`

(*wskTab)[2][1]

`wskTab[0][2][1]`
.....
`*(*(wskTab+2)+1)`

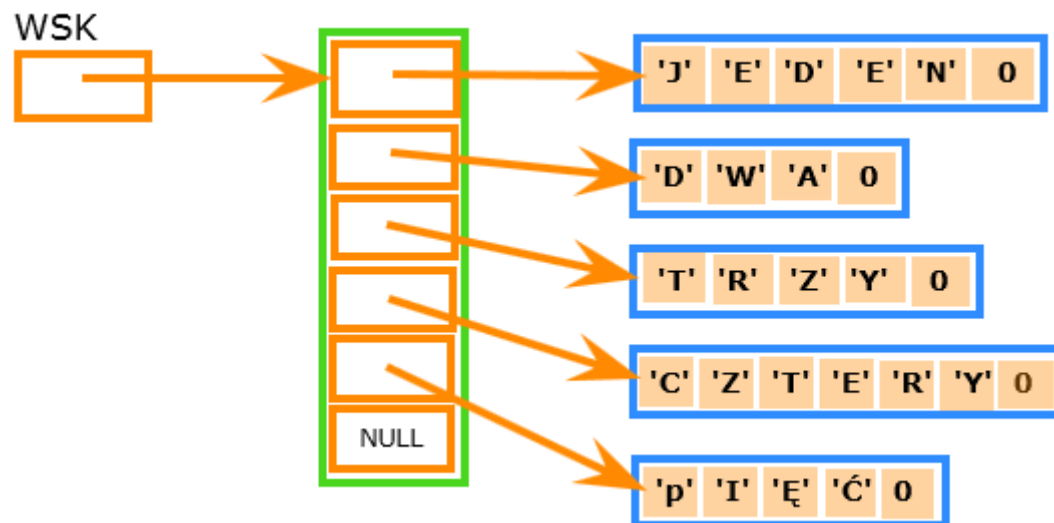
Dwuwymiarowa tablica „dynamiczna”

```
char **wskT;
```



Dwuwymiarowa tablica „dynamiczna”

```
char **WSK;
```

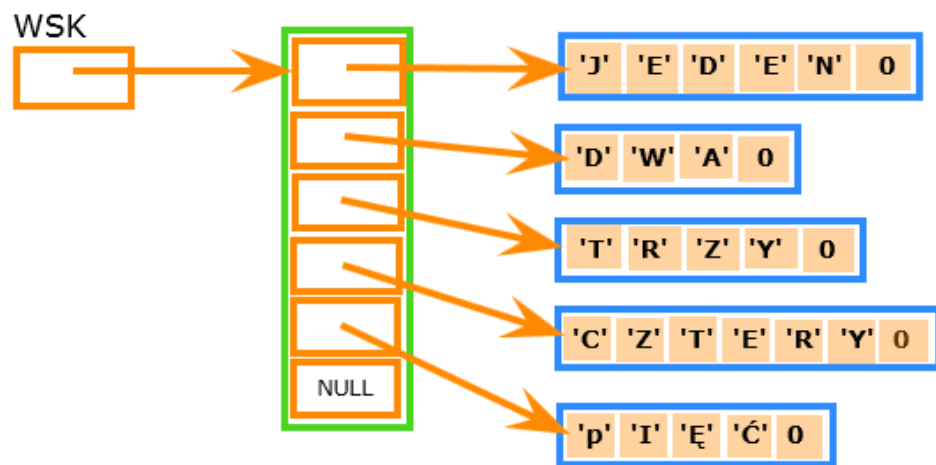


Dwuwymiarowa tablica „dynamiczna”

Tablica wskaźników oraz wszystkie tablice znaków są zmiennymi dynamicznymi (tzn. są tworzone poprzez alokację pamięci i nie posiadają swojej nazwy)

Ta struktura danych tworzy dynamiczną dwuwymiarową tablicę znaków.

```
char **WSK;
```

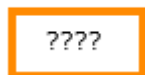


Wskaźnik `WSK` nie określa liczby wierszy i liczby kolumn w każdym wierszu. Dlatego ostatni element tablicy wskaźników musi mieć przypisaną wartość `NULL`, a każdy łańcuch znaków musi kończyć się znakiem o kodzie ASCII równym 0. To pozwala sprawdzić aktualną liczbę wierszy oraz liczbę kolumn w każdym wierszu.

Dwuwymiarowa tablica „dynamiczna”

Stan przed inicjalizacją:

WSK



1) Inicjalizacja tablicy dynamicznej

```
// utworzenie tablicy wskaźników  
// zawierającej 1 element  
WSK = (char **)malloc(sizeof(char *));  
WSK[0] = NULL;
```

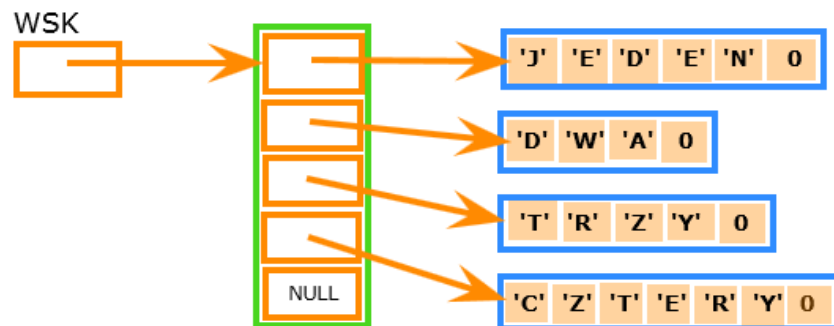
Stan po inicjalizacji:

WSK



Dwuwymiarowa tablica „dynamiczna”

Stan przed dodaniem:



2) Dodanie do tablicy nowego wiersza

```
char buf[81]; // tablica znaków, w której  
              // jest dodawany tekst
```

```
int poz = 0; // pozycja zawierająca NULL  
while( WSK[poz] != NULL ) poz++;
```

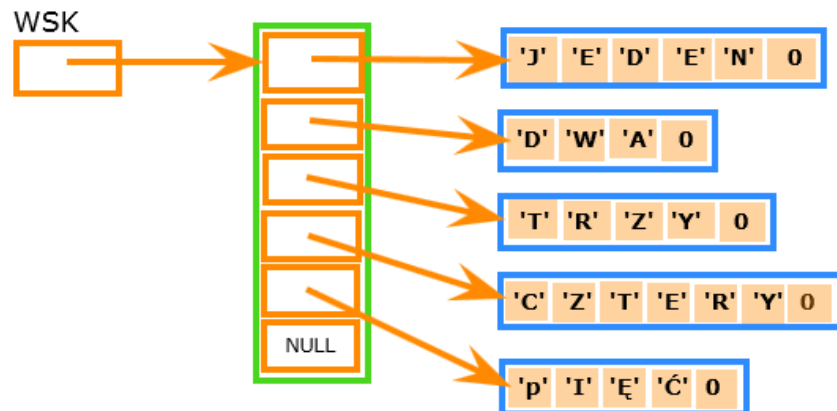
```
// powiększenie tablicy wskaźników  
WSK = (char**)realloc(WSK. (poz+2)*sizeof(char *));  
WSK[poz+1] = NULL;
```

```
// skopiowanie tekstu z bufora  
WSK[poz] = strdup(buf);
```

UWAGA: zamiast strdup można użyć malloc i strcpy:

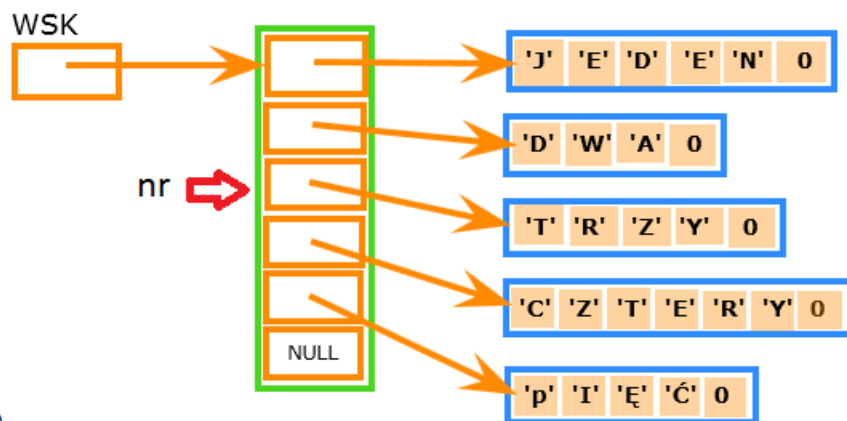
```
WSK[poz]=  
    (char*)malloc(sizeof(char)*(strlen(buf)+1);  
strcpy(WSK[poz], buf);
```

Stan po dodaniu:

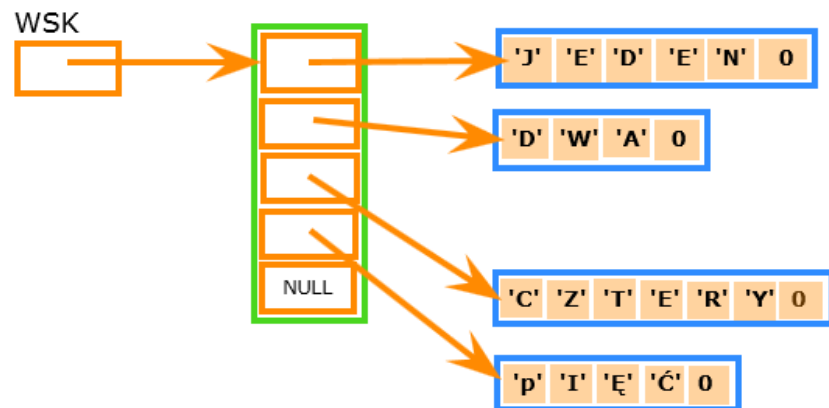


Dwuwymiarowa tablica „dynamiczna”

Stan przed usuwaniem:



Stan po usuwaniu:



3) Usuwanie z tablicy wskazanego wiersza

```
int nr ; //indeks wiersza, z którego
        // jest usuwany tekst
```

```
int poz = 0; // pozycja zawierająca NULL
while( WSK[poz] != NULL ) poz++;
```

```
if (nr >= poz) { // NIE WOLNO !!!
                return;
            }
```

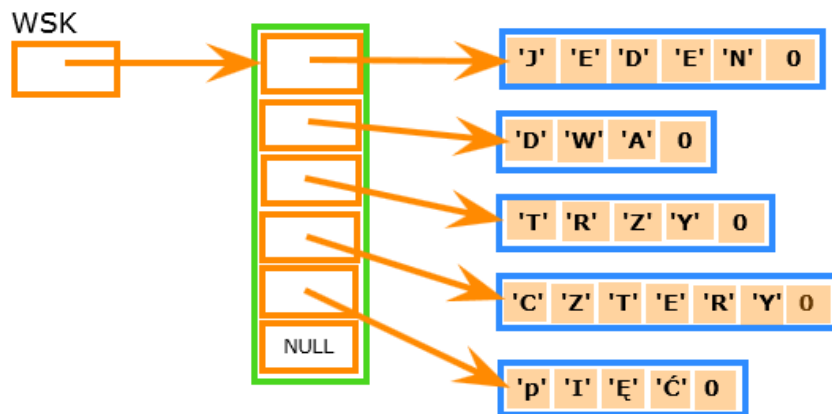
```
// zwolnienie pamięci
free(WSK[nr]);
```

```
// zsunięcie elementów tablicy
for(int i=nr; i<poz; i++) WSK[i] = WSK[i+1];
```

```
// pomniejszenie tablicy wskaźników
WSK = (char**)realloc(WSK. (poz*sizeof(char *));
```


Dwuwymiarowa tablica „dynamiczna”

Stan przed sortowaniem:



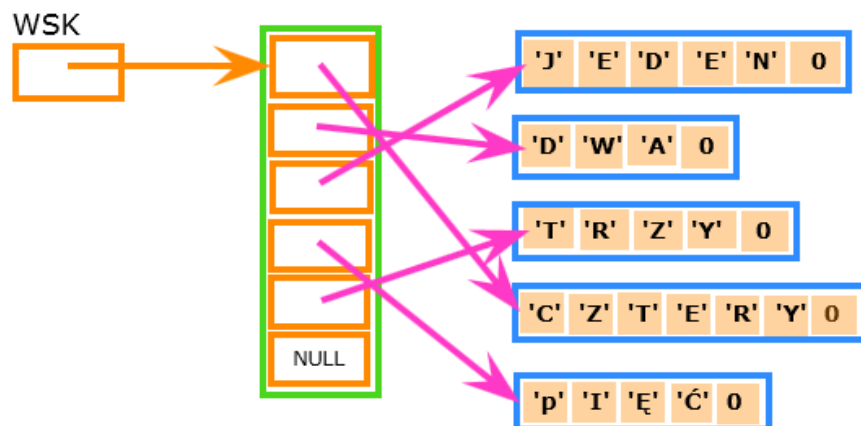
4) Sortowanie tablicy wg alfabetu

UWAGA:

podczas sortowania nie należy zmieniać miejsca alokacji poszczególnych wierszy.

Sortowanie polega na uporządkowaniu tablicy wskaźników, tak by kolejne elementy wskazywały łańcuchy w kolejności alfabetycznej.

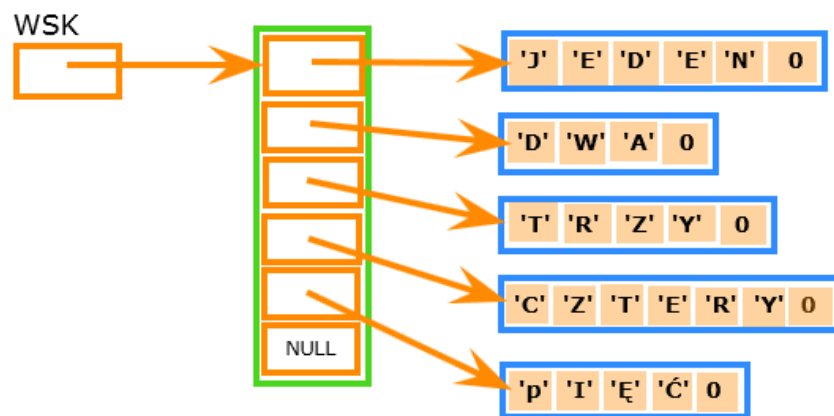
Stan po sortowaniu:



Do porównywania łańcuchów można wykorzystać np. funkcję `strcmp`

Dwuwymiarowa tablica „dynamiczna”

Stan przed sortowaniem:



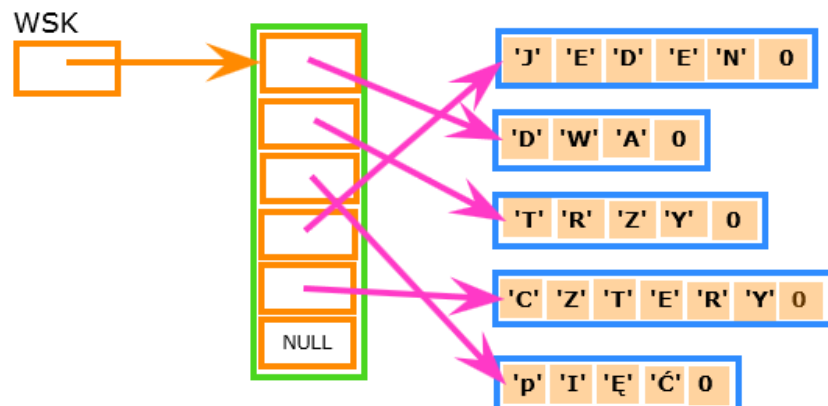
4) Sortowanie tablicy wg długości tekstu

UWAGA:

podczas sortowania nie należy zmieniać miejsca alokacji poszczególnych wierszy.

Sortowanie polega na uporządkowaniu tablicy wskaźników, tak by kolejne elementy wskazywały łańcuchy w kolejności od najkrótszego do najdłuższego.

Stan po sortowaniu:



Do porównywania długości można wykorzystać funkcję `strlen`