

## Przykłady dynamicznego tworzenia i usuwania różnych wariantów tablicy dwuwymiarowej → int [2][3]

**Przykład A)** dla porównania → zwykła tablica dwuwymiarowa

```
void main( )
{
int tab_A [ 2 ] [ 3 ];
    // przykładowe operacje
    tab_A[0][0] = 200;
    tab_A[0][1] = 201;
    tab_A[1][0] = 210;
    tab_A[1][1] = tab_A[1][2];
    // nie trzeba oprogramowywać operacji usuwania tablicy A
    // bo jej tworzenie i usuwanie jest realizowane przez kompilator !
}
```

**Przykład B)**

Wskaźnik na dwuwymiarową dynamiczną tablicę liczb całkowitych

Uwaga: taka reprezentacja jest bardzo niewygodna w zapisie,  
bo wszędzie trzeba dopisywać operator „gwiazdki” lub [0]

```
void main( )
{
int (*tab_B) [ 2 ] [ 3 ];
    tab_B = new int[1][2][3];           // dynamiczna alokacja tablicy tab_B
    // przykładowe operacje na takiej tablicy B
    (*tab_B)[0][0] = 200;               // tab_B[0][0][0] = 200;
    (*tab_B)[0][1] = 201;               // tab_B[0][0][1] = 201;
    (*tab_B)[1][0] = 210;               // tab_B[0][1][0] = 210;
    (*tab_B)[1][1] = (*tab_B)[1][2];    // tab_B[0][1][1] = tab_B[0][1][2];
    delete [ ] tab_B;                   // zwolnienie pamięci
}
```

```
#include <alloc.h>    // to samo, za pomocą tradycyjnych funkcji malloc i free
tab_B = ( int(*)[2][3] )malloc( 2*3*sizeof(int) ); // utworzenie
(*tab_B)[0][0] = 200; // tab_B[0][0][0] = 200;
free( tab_B );        // zwolnienie
```

## Przykład C)

Wskaźnik na pierwszą 3-elementową tablicę (pierwszą z dwóch)

```
void main( )
{
int (*tab_C) [ 3 ];
    tab_C = new int [2][3];           // dynamiczna alokacja tablicy tab_C
    tab_C[0][0] = 200;                // przykładowe operacje na tablicy tab_C
    tab_C[0][1] = 201;
    tab_C[1][0] = 210;
    tab_C[1][1] = tab_C[1][2];
    delete [ ] tab_C;                 // zwolnienie pamięci tab_C
}
```

```
#include <alloc.h>    // to samo, za pomocą tradycyjnych funkcji malloc i free
tab_C = ( int(*)[3] ) malloc( 2*3*sizeof(int) );           // utworzenie
tab_C[0][0] = 200;
free( tab_C );                                           // zwolnienie
```

## Przykład D)

Zwykła dwuelementowa tablica wskaźników na dynamiczne tablice liczb

```
void main( )
{
int *tab_D [ 2 ];
    tab_D[0] = new int [3];           // pierwszy wiersz ← dynamiczna alokacja
    tab_D[1] = new int [3];           // drugi wiersz
    tab_D[0][0] = 200;                // przykładowe operacje na tablicy tab_D
    tab_D[0][1] = 201;
    tab_D[1][0] = 210;
    tab_D[1][1] = tab_D[1][2];
    delete [ ] tab_D[0];              // pierwszy wiersz ← zwolnienie pamięci
    delete [ ] tab_D[1];              // drugi wiersz
}
```

```
#include <alloc.h>    // to samo, za pomocą tradycyjnych funkcji malloc i free
tab_D[0] = (int*)malloc(3*sizeof(int));  tab_D[1] = (int*)malloc(3*sizeof(int));
tab_D[0][0] = 200;
free( tab_D[0] );  free( tab_D[1] );
```

## Przykład E)

**Wskaźnik na dynamiczną (dwuelementową) tablicę wskaźników,  
na dynamiczne (trzyelementowe) tablice liczb całkowitych**

```
void main( )
{
int **tab_E;
// Dwuetapowe tworzenie tablicy -> najpierw tablica z adresami wierszy
tab_E = new int* [2];           // tablica dwóch wskaźników
// dopiero potem, dwie tablice zawierające wiersze
tab_E[0] = new int [3];        // pierwszy wiersz
tab_E[1] = new int [3];        // drugi wiersz

tab_E[0][0] = 200;             // przykładowe operacje na tablicy E
tab_E[0][1] = 201;
tab_E[1][0] = 210;
tab_E[1][1] = tab_E[1][2];

// zwolnienie obszaru zajmowanego przez tablicę E
delete [ ] tab_E[0];          // zwolnienie pierwszego wiersza
delete [ ] tab_E[1];          // zwolnienie drugiego wiersza
delete [ ] tab_E ;           // dopiero na koniec, zwolnienie tablicy wskaźników
}
```

```
// zazwyczaj operacja alokowania i zwalniania wierszy wykonywana jest w pętli:
for( int i=0; i<ILOSC_WIERWSZY; i++)
    delete [ ] tab_E[ i ];    // zwolnienie i-tego wiersza
delete [ ] tab_E ;           // dopiero na koniec, zwolnienie tablicy wskaźników
```

```
#include <alloc.h>           // to samo, za pomocą tradycyjnych funkcji malloc i free
tab_E = (int**)malloc( 2*sizeof(int*) ); // tworzenie tablicy wskaźników
tab_E[0] = (int*)malloc( 3*sizeof(int) ); // pierwszy wiersz
tab_E[1] = (int*)malloc( 3*sizeof(int) ); // drugi wiersz

tab_E[0][0] = 200;           // przykładowe operacje na tablicy E
// zwolnienie obszaru zajmowanego przez tablicę E
free( tab_E[0] );           // zwolnienie pierwszego wiersza
free( tab_E[1] );           // zwolnienie drugiego wiersza
free( tab_E );              // dopiero na koniec, zwolnienie tablicy wskaźników
```