// PROGRAM 4_1 - Przyklady dynamicznego tworzenia
// i usuwania tablicy dwuwymiarowej int [2][3]

```
#include <alloc.h>

void main(void)
{
```

//---------- Przyklad A ----------
//zwykla tablica dwuwymiarowa (dla porównania)

```
int tab_A [ 2 ] [ 3 ];

// przykladowe operacje
tab_A[0][0] = 200;
tab_A[0][1] = 201;
tab_A[1][0] = 210;
tab_A[1][1] = tab_A[1][1];

// nie trzeba oprogramowywac operacji usuwania tablicy A
// bo jej tworzenie i usuwanie jest zrealizowane przez kompilator !
```

//---------- Przyklad B ----------
// wskaznik na dwuwymiarowa dynamiczna tablice liczb calkowitych
// Uwaga: taka reprezentacja jest bardzo niewygodna do zapisywania
// bo wszedzie trzeba dopisywac operator „gwiazdki” lub [0]

```
int (*tab_B) [ 2 ] [ 3 ];

tab_B = ( int(*)[2][3] )malloc( 2*3*sizeof(int) );
// ## tab_B = new int[1][2][3];    << to samo, za pomoca operatora „new”

// przykladowe 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][1];

// zwolnienie obszaru zajmowanego przez tablice B
free( tab_B );
// ## delete [ ] tab_B;       << to samo, za pomoca operatora „delete”
```

//---------- Przyklad C ----------
//wskaznik na pierwsza 3-elementowa tablice (pierwsza z dwóch)

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

```
tab_C = ( int(*)[3] )malloc( 2*3*sizeof(int) );
// ##  tab_C = new int [2][3];

// przykladowe operacje na tablicy C
tab_C[0][0] = 200;
tab_C[0][1] = 201;
tab_C[1][0] = 210;
tab_C[1][1] = tab_C[1][1];

// zwolnienie obszaru zajmowanego przez tablice C
free( tab_C );
// ##  delete [ ] tab_C;
```

//---------- Przyklad D ----------
//zwykla dwuelementowa tablica wskazników na dynamiczne tablice liczb

```
int *tab_D [ 2 ];
```

```
tab_D[0] = (int*)malloc( 3*sizeof(int) );  // pierwszy wiersz
tab_D[1] = (int*)malloc( 3*sizeof(int) );  // drugi wiersz

// ##  tab_D[0] = new int [3];  // pierwszy wiersz
// ##  tab_D[1] = new int [3];  // drugi wiersz

// przykladowe operacje na tablicy D
tab_D[0][0] = 200;
tab_D[0][1] = 201;
tab_D[1][0] = 210;
tab_D[1][1] = tab_D[1][1];

// zwolnienie obszaru zajmowanego przez tablice D
free( tab_D[0] );
free( tab_D[1] );

// ##  delete [ ] tab_D[0];
// ##  delete [ ] tab_D[1];
```

//---------- Przyklad E ----------
// dynamiczna dwuelementowa tablica wskazników na dynamiczne tablice
// 3 liczb calkowitych

```
int **tab_E;

// Dwuetapowe tworzenie tablicy -> najpierw tablica z adresami wierszy
tab_E = (int**)malloc( 2*sizeof(int*) );     // tablica dwóch wskazników

// potem dwie tablice zawierajace wiersze
tab_E[0] = (int*)malloc( 3*sizeof(int) );    // pierwszy wiersz
tab_E[1] = (int*)malloc( 3*sizeof(int) );    // drugi wiersz

// ##  tab_E = new int* [2];       // tablica dwóch wskazników
// ##  tab_E[0] = new int [3];     // pierwszy wiersz
// ##  tab_E[1] = new int [3];     // drugi wiersz

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

// zwolnienie obszaru zajmowanego przez tablice E
free( tab_E[0] );     // zwolnienie pierwszego wiersza
free( tab_E[1] );     // zwolnienie drugiego wiersza
free( tab_E );        // dopiero na koniec, zwolnienie tablicy wskazników

// ##  delete [ ] tab_E[0];
// ##  delete [ ] tab_E[1];
// ##  delete [ ] tab_E ;

} // main – program 4_1
```