

## Homework 2

1. Implement the function `void v_alloc_table_add_5 (int iSize)`, which allocates in dynamic way memory for one-dimensional array of int variables. The size of an array is specified by the (iSize) parameter. The array elements should be initiated on offset+5.

- When array is allocated and initialized display all array elements.
- Remember that you need to deallocate memory using delete before end of the „program“.
- The function should be protected against an invalid iSize parameter value.
- Finally, the question, should the value 5 appear directly in the code of `v_alloc_table_add_5` function? Discuss it.

2. Implement the function

`bool b_alloc_table_2_dim (int ??? piTable, int iSizeX, int iSizeY)` for which:

- The allocation should be made in such way that after execution of the following code

```
int ** pi_table;  
b_alloc_table_2_dim (??? pi_table, 5, 3)
```

`pi_table` should pointed an int array of 5 \* 3 size.

- If the operation succeeds, the function should return true and false otherwise.
- Determine what to insert instead of `???` when the reference cannot be used.

3. Implement the function

`bool b_dealloc_table_2_dim (int ??? piTable, int iSizeX, int iSizeY);` which

- Deallocates a two-dimensional array of int type.
- If the operation succeeds, the function should return `true` and `false` otherwise.
- Determine what to insert instead of `???` when the reference cannot be used.
- Will there be a difference compared with function `b_alloc_table_2_dim`?
- Can `b_dealloc_table_2_dim` have fewer parameters then function `b_alloc_table_2_dim`?