

COMPUTER SCIENCE I

Exercise 8

1. Create a program in the following way:
 - In the main function declare a static two-dimensional array with M rows and N columns.
 - Write a function which will calculate the value of the elements according to the formula (i, j are indices of the element): $B_{ij} = (i+1)*(j+1)$.
 - Write a function printing the array.
 - Write a function which calculates average of all the elements of the array. The result should be printed inside the main function.
 - Write two functions which returns number of the elements greater than average and less than average. The result should be printed inside the main function.
 - Write a function which returns the n-th row of the array. The result should be passed to the function as an argument. Use for this purpose a one dimensional vector dynamically allocated in the main function.
 - Write a function which prints the n-th row obtained by using the function from previous point.
2. Modify the program by applying dynamic allocation to the array:

```
// allocate continuous block of the memory for all elements
double* p = (double* ) malloc( M * N * sizeof(double) );
// allocate memory for the vector storing pointers to the rows
double** tt = ( double** ) malloc ( M * sizeof(double) );
// assign proper addresses to the pointers from vector tt
for( int ii = 0; ii < M; ii++ )
{
    tt[ii] = p;
    p += NN; // shift pointer to the address of the next row
}

// ...write here a code which uses the array

// free memory
free( tt[0] ); // free memory used by the elements
free( tt ); // free memory used by the pointer vector
```

The address of the memory allocated for elements was assigned to the pointer p but later the pointer was shifted and it does not point to the block of memory anymore. The proper address is stored in the stored in the tt[0] and it can be used for freeing the allocated memory.