Note: For every exercise, create an own project. Exercises which are not finished in class must be completed at home!

Learning outcome of this set of exercises: Functions, parameters, pointers, header and source files, using the debugger.

Exercise categories:

- A – very basic, intended for inexperienced developers
- B – fair, a little bit more complex but still for starters
- C – challenging, complexity is higher, additional programming constructs may be required

Exercise 1 – Memory overflow (A)

Write a function which

- allocates a block of memory (size passed as parameter)
- prints the address and size of the allocated memory
- and releases it afterwards

Write a loop 1…100 and allocate “loop variable x 1kByte” using the previously defined function.

- Check the output
- Comment out the free command and run the program again – what happens
- Replace the for loop with a while(1) loop – what happens now?

Example output:

```
Allocated 1024 bytes at 552b08
Allocated 2048 bytes at 552b08
Allocated 3072 bytes at 552b08
Allocated 4096 bytes at 552b08
Allocated 5120 bytes at 552b08
Allocated 6144 bytes at 552b08
Allocated 7168 bytes at 552b08
Allocated 8192 bytes at 552b08
Allocated 9216 bytes at 552b08

Allocated 10240 bytes at 552b08
```

Exercise 2 – Board game (B)

Write a small program which serves as a starting point for a board game:

- The board is a char array with a size of x columns and y rows
- The size of the board is entered by the user
- Once the board is created, the user can enter characters at any position
- The board can be printed on the screen

Hints:

- C does not provide 2d char arrays. The 2d array is represented by an x*y byte memory.
- Write 3 functions clearBoard, printBoard and setBoard, taking a pointer to the board and additional parameters as needed.
- Test the program after every function

Example output:

```
The starting point of a board game
Please enter the size (x,y) of the game:
3 5
Allocated board of dimension x = 3, y = 5 at adress 672b08
. . .
. . .
. . .
. . .
. . .
Please enter a position (x,y) and a stone character (0 to end):
2 4 x
. . .
. . .
. . .
. . .
. . x
Please enter a position (x,y) and a stone character (0 to end):
1 3 o
. . .
. . .
. . .
. o .
. . x
Please enter a position (x,y) and a stone character (0 to end):
```