Programming – 2019-11-26

* “Yesterday we looked at the Malloc function” (refer to previous, yesterdays, notes).
* **EXAM NEXT WEEK WILL NOT BE ON DMA**
* “Calloc()” stands for Contiguous Allocation
* () = Parameters
* (int\*)malloc(no\_bytes) = where the (int\*) is a “cast”.
* *Note: Cast was not described in this class, seems it might be explained later.*
* Calloc automatically initialises the block to contain zeroes, rather then random data.
* Calloc also contains two data points, number\_of\_data\_items and size\_of\_single\_data\_item.
* EX1.c – Explanation of Calloc
* Calloc is a more efficient use of Malloc, even though in this program it is only one line less. In programs containing 1000’s of lines of code, the effect is cumulative.
* The M in malloc() stands for Memory
* “You are the designer of a program and are asked to enter in a set of numbers **while the program is running**, then calculate a set of numbers”.
* This caught many out, as they used an array. As an array is defined at compiletime, it cannot be done in runtime. Pointers, calloc() and malloc() can however be done in runtime. **LOOK FOR THE “WHILE THE PROGRAM IS RUNNING” part.**
* TL;DR: You cannot do an array at runtime, you can with DMA
* realloc() : calloc() : malloc()
* The RE in realloc() stands for reallocate, it will ONLY reallocate the memory block. It changes the size of the memory block later in the program, after initially defined. It will however not change the operator type.
* Pointer = realloc(pointer, new.size);
* TL;DR: It changes the size of a block.
* Realloc allows you to reduce the size of a bloc, not just increase it.
* Realloc will just extend the end of a memory block, it won’t move it. This means extending a memory block with realloc is a risky prospect.
* If you return a NULL in a realloc, you loose that ENTIRE block.