Local arrays are created on stack and their lifetime is only as long as you are within the function. Once you leave the function, you no longer own that memory area on stack, and it is reused by some other function that you call.

**Whenever you enter a function or a new scope, a stack frame gets creates for local variables and it gets destroyed the moment you leave the function or scope.**

If you want to prepare a buffer in a function, you must acquire a memory area that can stay alive even after you exit the function. This is where dynamic memory helps us. Allocate the memory you need on heap, fill it up with relevant data and return address of this memory area. Heap memory stays alive until you explicitly free it. Or you can create that array in the calling function and pass its pointer to called function.