1. I think that because there may be huge amounts parameters there needs to be a way for each to be contained separately. Since order matters a stack would work best.
2. Passing by value: 1, 10, 11
   Passing by reference: 3, 2, 11
   Passing by name: 3, 10, 2
3. The result of the code is all 0s or “0 1 2 3 4 5 6 7 8 9”. This is because that everytime the function foo() is called again it is always hitting the “int i;” command which either initialize it to 0 or pulls from the old “garbage” depending on the OS. I tried this out on two different OS online C-Compilers. The one that initializes the i to null then adds to 0 each time continues to use a new memory location as it should be doing. But the one that increases its value each time is reusing an old memory allocation and reusing what was last in the value as its initialized value.