Input: Function call *code* in function *func* at WyIL level
Output: A *list* of actual function call code

```
1: Variables
   CopyAnalyser: Copy elimination analyser
   DeallocAnalyser: De-allocation analyser
   VarStore: A variable set stores the names of temporary variables
2: end Variables
3: // Optimise actual function call
4: procedure OPTIMISE_ACTUALFUNCTIONCALL(code, func)
5:     list = []
6:     callee_name = get the name of called function at code
7:     ret = function return variable at code
8:     // Beginning of a function call
9:     if ret is NOT NULL then
10:        list.append(" ret = callee_name( ")
11:    else
12:        list.append(" callee_name( ")
13:    end if
14:    for index ← 0; index < |params|; index++ do
15:        param = params[index]
16:        if param is an Array then
17:            param_size = get size of array param
18:            if CopyAnalyser.isCopyRemoved(param, code, func) then
19:                // Pass param with false flag
20:                list.append(" param, param_size, false ")
21:            else
22:                macro ←DeallocAnalyser.choosePostMacro(param,
                       code, func)
23:                tmp ←VarStore(index, code, func)// Temporary variable
24:                // Pass copied parameter tmp
25:                if macro == CALLEE then
26:                    list.append(" tmp, param_size, true ")
27:                else// CALLER
28:                    list.append(" tmp, param_size, false ")
29:                end if
30:            end if
31:        else
32:            list.append(" param ")// Pass param without optimisation
33:        end if
34:    end for
35:    list.append(" ); ")// Ending of a function call
36:    return list
37: end procedure
```