

Input: One line of code in function *func*

Output: A list of post-deallocation *macros* suggested for *code* in *func*

```
1: Variables
2:   MutateAnalyser: Read/Write Analyser
3:   ReturnAnalyser: Return analyser
4:   LiveAnalyser: Live variable analyser
5:   CopyAnalyser: Copy elimination analyser
6:   macros: A list of macros used in code
7:   aParam: Actual parameter used in function call code
8:   fParam: Formal parameter used in definition of called function
9: end Variables
10: procedure COMPUTEPOSTDEALLOCMACRO(code, func)
11:   macros = [] // Store all macros used in code
12:   if code is ArrayGenerator then
13:     target = array variable of code
14:     macros.append(NEWARR_DEALLOC(target))
15:   else if code is Assignment then
16:     lhs = left-handed side of code
17:     rhs = right-handed side of code
18:     if lhs is an Array AND
19:       CopyAnalyser.isCopyRemoved(rhs, code, func) then
20:         macros.append(TRANSFER_DEALLOC(lhs, rhs))
21:       else // Copy of rhs is NOT removed at code
22:         macros.append(ADD_DEALLOC(lhs, rhs))
23:       end if
24:   else if code is Function call then
25:     ret = return variable of code
26:     callee = called function of code
27:     for each aParam in code do // Iterate each actual parameter aParam
28:       if aParam is Array then // Macro is applied on array type only
29:         isLive  $\leftarrow$  LiveAnalyser.isLive(aParam, func)
30:         fParam = map aParam at caller func to
31:           formal parameter at called function callee
32:         isMutated  $\leftarrow$  MutateAnalyser.isMutated(fParam, callee)
33:         isReturned  $\leftarrow$  ReturnAnalyser.isReturn(fParam, callee)
34:         switch isMutated – isReturned – isLive do
35:           case F-F-T  $\vee$  F-F-F  $\vee$  T-F-F
36:             macros.append(RETAIN_DEALLOC(ret, aParam))
37:           case F-T-F  $\vee$  T-T-F
38:             macros.append(RESET_DEALLOC(ret, aParam))
39:           case F-T-T  $\vee$  T-T-T
40:             macros.append(CALLER_DEALLOC(ret, aParam))
41:           case T-F-T
42:             macros.append(CALLEE_DEALLOC(ret, aParam))
43:         end if
44:       end for
45:   else // No needs to use post-deallocation macro
46:   end if
47: end procedure
```