**splint.c: (in function one)**

splint.c:9:7: Test expression for if is assignment expression: temp = *A
  The condition test is an assignment expression. Probably, you mean to use ==
  instead of =. If an assignment is intended, add an extra parentheses nesting
  (e.g., if ((a = b)) ...) to suppress this message. (Use -predassign to
  inhibit warning)

splint.c:9:7: Test expression for if not boolean, type int: temp = *A
  Test expression type is not boolean or int. (Use -predboolint to inhibit
  warning)

splint.c:11:15: Stack-allocated storage ptr reachable from global ptr
  A stack reference is pointed to by an external reference when the function
  returns. The stack-allocated storage is destroyed after the call, leaving a
  dangling reference. (Use -stackref to inhibit warning)

splint.c:10:3: Storage ptr becomes stack-allocated storage

**splint.c: (in function two)**

splint.c:15:12: Dereference of possibly null pointer [result of malloc]:
  *(int *)malloc(sizeof(int))
  A possibly null pointer is dereferenced. Value is either the result of a
  function which may return null (in which case, code should check it is not
  null), or a global, parameter or structure field declared with the null
  qualifier. (Use -nullderef to inhibit warning)

splint.c:15:11: Value used before definition
  An rvalue is used that may not be initialized to a value on some execution
  path. (Use -usedef to inhibit warning)

**splint.c: (in function three)**

splint.c:24:7: Test expression for if not boolean, type int: b % 2

splint.c:25:5: Variable k used before definition

splint.c:26:10: Unqualified storage ptr passed as only param: free (ptr)
  Unqualified storage is transferred in an inconsistent way. (Use
  -unqualifiedtrans to inhibit warning)
  splint.c:10:3: Storage ptr becomes stack-allocated storage

splint.c:29:10: Variable k used before definition

splint.c:29:12: Function returns with global ptr referencing released storage
  A global variable does not satisfy its annotations when control is
  transferred. (Use -globstate to inhibit warning)
  splint.c:26:10: Storage ptr released
splint.c: (in function four)
splint.c:35:15: Operands of < have incompatible types (int, size_t):  
i < sizeof((A))  
   To allow arbitrary integral types to match any integral type, use  
   +matchanyintegral.

splint.c:44:14: Index of possibly null pointer p: p
   splint.c:33:12: Storage p may become null

splint.c:44:14: Value p[] used before definition

splint.c:45:3: Variable p is released in true branch, but live in false branch.  
The state of a variable is different depending on which branch is taken. This  
means no annotation can sensibly be applied to the storage. (Use -branchstate  
to inhibit warning)
   splint.c:45:3: in true branch:
   splint.c:41:24: Storage p released
   splint.c:45:3: in false branch:
   splint.c:33:39: Storage p allocated

splint.c:4:6: Variable exported but not used outside splint: ptr  
A declaration is exported, but not used outside this module. Declaration can  
use static qualifier. (Use -exportlocal to inhibit warning)