

Appendix C Error Messages

- 1 or 2 SYNTAX ERR Several errors can cause this message.
- 3 MISSING LINE A reference was made to a non-existent line.
- 4 MISSING QUOTE Quotes missing from a string expression.
- 5 ARGUMENT ERR Several errors can cause this message. You might have forgotten to dimension an array or used the DIM statement after the first use of an array. The arguments in a function or named subroutine may differ from the way the function or subroutine was defined. Remember, in user-defined functions and subroutines, the system will not convert between integers and real numbers.
- 6 UNDEF.FN,SUB,OR ARRAY You forgot to define a function or subroutine or to dimension an array. Another possibility is a spelling mistake.
- 7 REDEF.FN,SUB,OR ARRAY You defined a function or subroutine twice, dimensioned an array twice, or used an array before the DIM statement.
- 11 MUST BE AT LINE START DATA,SUB, and DEF statements must be first in a line; the statement 100 A%=Z%: DATA 5 gives this error.
- 12 MAX.NUMB.EXC. Maximum number of variables (255) or maximum number of lines (975) exceeded. If maximum number of variables exceeded, I suggest you try executing CLEAN.COD to remove unused variables, if any.
- 13 ARITHMETIC ERR. Possibilities include you divided by zero, or tried to create a floating point number larger than or equal to 10^{99} or smaller or equal in absolute value to 10^{-99} .
- 14 EXP.TOO COMPLEX There is a maximum level of complexity possible in arithmetic expressions and a limit to the depth of nesting in loops. An error in one or both of these areas could cause this message.
- 15 STACK ERR. Possibilities include using POPST without LOADST or vice versa, thus exceeding the stack limit.
- 16 OUT OF DATA You have tried to read more data than is present in the program.
- 17 MEMORY EXCEEDED See Chapter 11.
- 18 NUMBER OUT OF RANGE The most likely cause is that you exceeded the maximum possible subscript value in an array. It is also possible that you used a negative subscript, or that in a string function, you used a number beyond the length of the string.
- 19 NO GOSUB Your program tried to execute a RETURN, but had never executed a GOSUB.
- 20 NO FOR You used a NEXT with no matching FOR. Note that this could be a spelling error. Unlike some BASICs, Advan checks the variable in the FOR and NEXT to make sure they agree.
- 21 NO CASE You used CASE END,CASE ELSE, or & with no CASE. All CASE