E

ANSI Header Files

The American National Standards Institute (ANSI) has set standards on many of the header files that are used in C. The following are header files that fall under the ANSI standard:

ASSERT.H

Contains the assert debugging macro.

CTYPE.H

Character classification and conversion macros that deal with single characters. Examples include:

```
isalnum()
isalpha()
iscntrl()
isdigit()
isgraph()
islower()
isprint()
ispunct()
isspace()
isupper()
isxdigit()
tolower()
toupper()
```

ERRNO.H

Defines constants for the error codes.

FLOAT.H

Defines limits and other constants for floating-point data types. Complements the limits.h header file.

LIMITS.H

Defines limits and other constants for integer variable types. Complements the float.h header file.

LOCALE.H

Defines country- and language-specific information. This includes information on currency format, numeric format, and date format.

math.h

Defines prototypes and constants used with math functions.

setjmp.h

Contains prototypes and other information used in the jump functions—`setjmp()` and `longjmp()`.

signal.h

Defines prototypes and constants that are used with the signaling functions—`signal()` and `raise()`.

stdarg.h

Defines macros and constants for functions that use a variable number of arguments.

stddef.h

Defines several common constants, identifiers, variables, and more.

stdio.h

Defines constants used with standard input and output. This includes the common stream: `stdin`, `stdout`, `stderr`, and `stdprn`.

stdlib.h

Defines prototypes and commonly used declarations for standard functions. This header contains those items that didn't really fit in any of the other ANSI header files.

string.h

Defines constants and prototypes for functions that work with character arrays (strings). This includes functions that work with memory.

time.h

Defines constants, structures, and prototypes for time-related functions.

Additionally…

In addition to these standard header files, your compiler has several of its own. Consult your compiler's reference manuals to see what header files and functions it has in addition to the standard ANSI header files. If you decide to use any of your compiler's non-ANSI header files, you may be limiting the portability of your programs. If you

E

only use the ANSI header files, then you'll know that your programs will be portable to any other ANSI compatible compiler.