C LIBRARY - <SIGNAL.H>

The **signal.h** header defines a variable type **sig_atomic_t**, two function calls, and several macros to handle different signals reported during a program's execution.

Library Variables

Following is the variable type defined in the header signal.h −

S.N.	Variable & Description
1	**sig_atomic_t** This is of **int** type and is used as a variable in a signal handler. This is an integral type of an object that can be accessed as an atomic entity, even in the presence of asynchronous signals.

Library Macros

Following are the macros defined in the header signal.h and these macros will be used in two functions listed below. The **SIG_** macros are used with the signal function to define signal functions.

S.N.	Macro & Description
1	**SIG_DFL** Default signal handler.
2	**SIG_ERR** Represents a signal error.
3	**SIG_IGN** Signal ignore.

The **SIG** macros are used to represent a signal number in the following conditions −

S.N.	Macro & Description
1	**SIGABRT** Abnormal program termination.
2	**SIGFPE** Floating-point error like division by zero.
3	**SIGILL** Illegal operation.

4 **SIGINT**

Interrupt signal such as ctrl-C.

5 **SIGSEGV**

Invalid access to storage like segment violation.

6 **SIGTERM**

Termination request.

Library Functions

Following are the functions defined in the header signal.h −

S.N.	Function & Description
1	void (*signal(int sig, void(*func)(int)))(int) This function sets a function to handle signal i.e. a signal handler.
2	int raise(int sig) This function causes signal **sig** to be generated. The sig argument is compatible with the SIG macros.