assert.h in C

The file header named assert.h from Library C provides a macro called assert that can be used to test an assumption made by the program and print a diagnostic message to find the error if this assumption is false.

assert.h in C

The file header named **assert.h** from Library C provides a macro called **assert** that can be used to test an assumption made by the program and print a diagnostic message to find the error if this assumption is **false** .

The macro is predefined **assert** this reference to another macro is **NDEBUG** which is not part of. If **NDEBUG** is defined as a macro name in the source file, where it is included, assert is defined as follows:

```
#define assert ( ignore ) (( void ) 0 )
```

The macros are defined in assert.h

In assert.h only define the following macro:

void assert (int expression): This is really a macro and not a function. This macro is used to add diagnostics to find errors in the C program.

Assert macro () in C

The assert () function in C

Macro void assert (int bieu-thuc) in Standard C Library allows detection diagnostic information to be written to standard error file (Standard Error File). In other words, it can be used to add diagnostic messages to find errors in C programming.

Declare assert () function in C

Below is the declaration for the assert () Macro in the standard C Library.

```
void assert ( int  bieu - thuc );
```

Parameters

bieu-thuc: This can be a variable or any C expression. If the command evaluates to TRUE, then assert () does nothing. If the method evaluates to false, then assert () displays an error message on stderr and stops executing the program.

Return value

This macro does not return any values.

For example

The following C program illustrates the use of assert () macros in Library C:

```
#include #include int  main () { int  a ; char  str [ 50 ];  printf ( "Nhap mot
```

Compile and run the above program to see the results:

According to Tutorialspoint

Previous post: Command line parameter in C

Next story: ctype.h in C