Overview

Programming Language: C Programming
Software Tool: Ubuntu OS (Linux)
Compiler: GCC

Objective:

The objective of sharing this project is to present the test bench by writing the unit tests that cover the corner cases/boundary conditions. Likewise, one can do stress testing and regression testing on a large scale.

This code was developed to implement the **<u>test bench for unit testing</u>** of the ***uint_to_binstr()*** function. The .zip file contains a makefile, uint_to_binstr.c, uint_to_binstr.h, unit_test.c, unit_test.h and main.c files.

<p align="center"><code>int uint_to_binstr(char *str, size_t size_str, uint32_t num, uint8_t nbits)</code></p>

Description:

This function returns the binary representation of an unsigned integer, as a null-terminated string. On input, *str* is a pointer to a char array of at least size bytes, *num* is the value to be converted, and *nbits* is the number of bits in the input.

If the operation is successful, the function returns the number of characters written to *str*, not including the terminal '\0'. In the case of an error, the function returns a negative value, and *str* is set to the empty string.

The test bench is developed to test the unit test cases of the program. The test bench covers various test cases including the "corner cases" or "boundary conditions" and logs the total number of passed cases out of the total of test cases.

The program was compiled using gcc with **–Wall** and **–Werror** flags to consider every warning as an error. As warnings are not considered good in programming. It is always recommended to resolve every warning.

Examples –

num	nbits	str	Return value
18	8	0b00010010	10
65400	16	0b1111111101111000	18
310	0	""	-1 (illegal nbits)
310	4	""	-1 (num too large)

Output –