**c\_unit assignment**

1. Extend the test code in “Example Code Snippets/min\_max\_app/Test” directory used in cunit demo to add test the following cases

a. getMin() with an array with all elements as 0

b. getMax() with an array with all elements as 0

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Running suite MIN\_MAX\_SUITE

Running test test\_getMin\_all\_zeroes

Running test test\_getMax\_all\_zeroes

All tests passed

2. Extend the code in “Example Code Snippets/min\_max\_app/src/min\_max.c, .h” to add a new function below to return the longest string. Also add unit test case to test the newly added function.

char \*getLongest(char \*s1, char \*s2)

[Consider all valid and invalid input cases]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* File Name:minmax\_test.c

\* Purpose: Main function to call the CUNIT Framework

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#include <sys/types.h

#include <Basic.h>

#include <CUnit.h>

#include <min\_max.h>

#include <minmax\_testfunctions.h>

extern void Mytestfunction\_min(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function : init\_minmax

Description : Initializes the suite

Input : Nothing

Output : int

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int init\_minmax(void)

{

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function : clean\_minmax

Description : clean the suite after successful run

Input : Nothing

Output : int

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int clean\_minmax(void)

{

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function : main

Description :This is the main function to test backend getMin(), getMax()

Input : None

Output : Int

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(int argc, char \*argv[])

{

CU\_pSuite pSuite = NULL;

/\* Register \*/

if (CUE\_SUCCESS != CU\_initialize\_registry())

return CU\_get\_error();

/\* Add init and cleanup handlers \*/

pSuite = CU\_add\_suite("LINKED LIST SUITE",init\_minmax,

clean\_minmax);

if (NULL == pSuite)

{

CU\_cleanup\_registry();

return CU\_get\_error();

}

/\* Add test case \*/

if(NULL == CU\_add\_test(pSuite,"Mytestfunction\_min\_ID\_1", Mytestfunction\_min))

{

CU\_cleanup\_registry();

return CU\_get\_error();

}

/\* Set the mode and invoke test(s) \*/

CU\_basic\_set\_mode(CU\_BRM\_VERBOSE);

CU\_basic\_run\_tests();

/\* unregister and do cleanup \*/

CU\_cleanup\_registry();

return CU\_get\_error();

}

A screenshot of a computer screen

Description automatically generated