1. [System Setup]

```
osboxes@osboxes:~/CSCE313$ g++ --version
g++ (Ubuntu 10.2.0-13ubuntu1) 10.2.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

osboxes@osboxes:~/CSCE313$ gdb --version
GNU gdb (Ubuntu 9.2-0ubuntu2) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
osboxes@osboxes:~/CSCE313$
```

2. [C++ Compilation]

```
 1 #include <iostream>
 2 #include <vector>
 3
 4 using namespace std;
 5  //Blank A
 6 class node {
 7 public:
 8  //Blank B
 9         int val;
10         node* next;
11 };
12
13 void create_LL(vector<node*>& mylist, int node_num){
14     mylist.assign(node_num, NULL);
15
16     //create a set of nodes
17     for (int i = 0; i < node_num; i++) {
18         //Blank C
19         mylist[i]->val = i;
20         mylist[i]->next = NULL;
21     }
22
23     //create a linked list
24     for (int i = 0; i < node_num; i++) {
25         mylist[i]->next = mylist[i+1];
26     }
27 }
28
29 int sum_LL(node* ptr) {
30     int ret = 0;
31     while(ptr) {
32         ret += ptr->val;
33         ptr = ptr->next;
34     }
35     return ret;
36 }
37
```

3. [Compilation with symbol table]

```
osboxes@osboxes:~/CSCE313$ g++ buggy.cpp -o buggy -g
```

4. [GDB Start/Run/Backtrace]

```
osboxes@osboxes:~/CSCE313$ g++ buggy.cpp -o buggy -g
osboxes@osboxes:~/CSCE313$ gdb buggy
GNU gdb (Ubuntu 9.2-0ubuntu2) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from buggy...
(gdb) run
Starting program: /home/osboxes/CSCE313/buggy

Program received signal SIGSEGV, Segmentation fault.
0x00005555555552f7 in create_LL (
    mylist=std::vector of length 3, capacity 3 = {...}, node_num=3)
    at buggy.cpp:19
19              mylist[i]->val = i;
(gdb) backtrace
#0  0x00005555555552f7 in create_LL (
    mylist=std::vector of length 3, capacity 3 = {...}, node_num=3)
    at buggy.cpp:19
#1  0x0000555555555406 in main (argc=1, argv=0x7fffffffe148) at buggy.cpp:42
```

5. [GDB Breakpoint/Print]

```
(gdb) b 19
Breakpoint 1 at 0x5555555552dc: file buggy.cpp, line 19.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/osboxes/CSCE313/buggy

Breakpoint 1, create_LL (mylist=std::vector of length 3, capacity 3 = {...}, node_num=3) at buggy.cpp:19
19              mylist[i]->val = i;
(gdb) print mylist[i]
$1 = (node *) 0x0
(gdb)
```

6. [C++ Runtime Error Fix (Null-Pointer)]

```
17      for (int i = 0; i < node_num; i++) {
18          //Blank C
19          mylist[i] = new node();
20          mylist[i]->val = i;
21          mylist[i]->next = NULL;
22      }
23
```

7. [C++ Runtime Error Fix (non-NULL garbage value Pointer)]

```
24      //create a linked list
25      for (int i = 0; i < node_num-1; i++) {
26          mylist[i]->next = mylist[i+1];
27      }
28  }
29
30  int sum_LL(node* ptr) {
31      int ret = 0;
32      while(ptr != null) {
33          ret += ptr->val;
34          ptr = ptr->next;
35      }
36      return ret;
37  }
```

Explanation:

In the sum_LL:

ptr is null if ptr is the value of node* part of the third node.

In the create_LL:

The loop end condition should be: i < node_num-1. If the end condition is: i < node_num, when i = node_num-1 = 2, mylist[i+1] = mylist[node_num] = mylist[3], which is not available(out of index).

8. [Deletion of Dynamically Allocated Memory]

```
47      //Step4: delete nodes
48      //Blank D
49      for (int i = 0; i < mylist.size(); i++) {
50          delete mylist[i];
51      }
```

9. [AddressSanitizer]

Compile the original file with AddressSanitizer:

```
osboxes@osboxes:~/CSCE313$ g++ buggy-original.cpp -o buggy-original -fsanitize=address
buggy-original.cpp:9:6: error: variable or field ‘create_LL’ declared void
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |      ^~~~~~~~~
buggy-original.cpp:9:16: error: ‘vector’ was not declared in this scope
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |                ^~~~~~
buggy-original.cpp:9:27: error: expected primary-expression before ‘*’ token
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |                           ^
buggy-original.cpp:9:28: error: expected primary-expression before ‘>’ token
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |                            ^
buggy-original.cpp:9:31: error: ‘mylist’ was not declared in this scope
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |                               ^~~~~~
buggy-original.cpp:9:39: error: expected primary-expression before ‘int’
    9 | void create_LL(vector<node*>& mylist, int node_num){
      |                                       ^~~
buggy-original.cpp: In function ‘int sum_LL(node*)’:
buggy-original.cpp:28:20: error: request for member ‘val’ in ‘ptr’, which is of pointer type ‘node*’ (maybe you meant to u
se ‘->’ ?)
   28 |         ret += ptr.val;
      |                    ^~~
buggy-original.cpp:29:19: error: request for member ‘next’ in ‘ptr’, which is of pointer type ‘node*’ (maybe you meant to
use ‘->’ ?)
   29 |         ptr = ptr.next;
      |                   ^~~~
buggy-original.cpp: In function ‘int main(int, char**)’:
buggy-original.cpp:36:5: error: ‘vector’ was not declared in this scope
   36 |     vector<node*> mylist;
      |     ^~~~~~
buggy-original.cpp:36:16: error: expected primary-expression before ‘*’ token
   36 |     vector<node*> mylist;
      |                ^
```

After fixing all bugs except deleting the allocated dynamically memory, using the AddressSanitizer, it can detect the memory leaks.

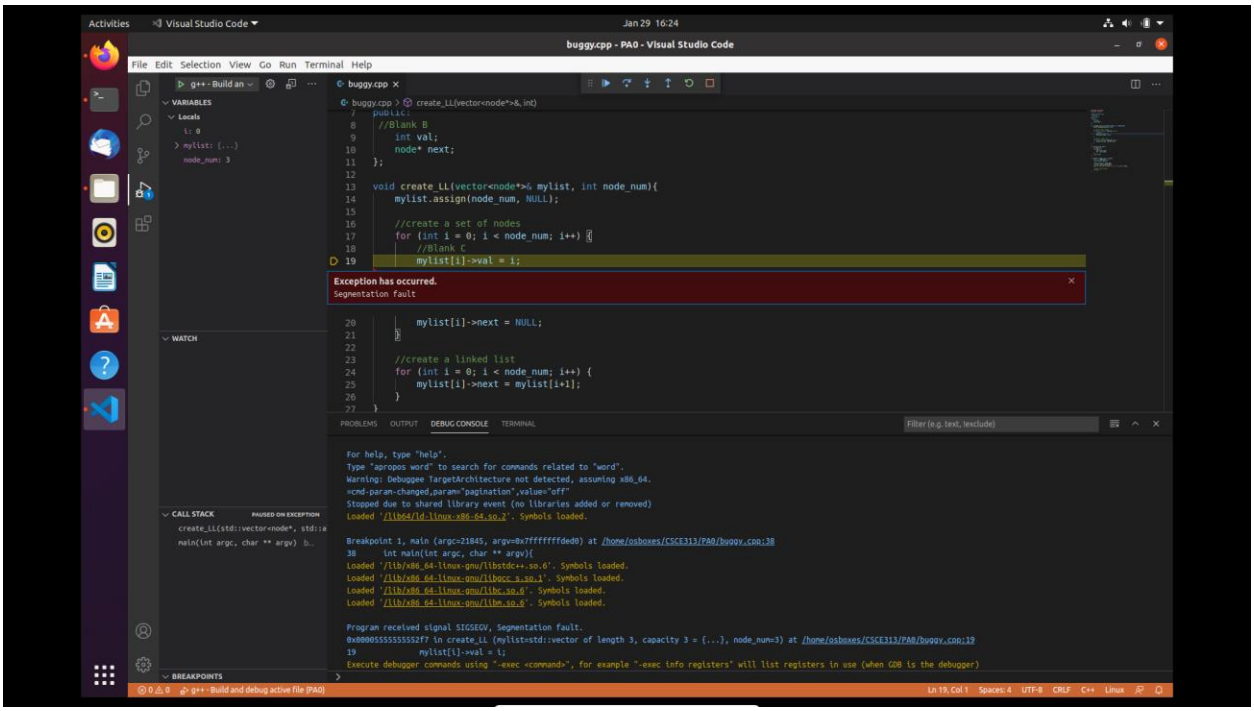
```
osboxes@osboxes:~/CSCE313$ g++ buggy-original.cpp -o buggy-original -fsanitize=address
osboxes@osboxes:~/CSCE313$ ./buggy-original
The sum of nodes in LL is 3

=================================================================
==52387==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 16 byte(s) in 1 object(s) allocated from:
    #0 0x7f738d113f17 in operator new(unsigned long) (/lib/x86_64-linux-gnu/libasan.so.6+0xb1f17)
    #1 0x55d88c48053b in create_LL(std::vector<node*, std::allocator<node*> >&, int) (/home/osboxes/CSCE313/buggy-original
+0x253b)
    #2 0x55d88c48094c in main (/home/osboxes/CSCE313/buggy-original+0x294c)
    #3 0x7f738cca4cb1 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x28cb1)

Indirect leak of 32 byte(s) in 2 object(s) allocated from:
    #0 0x7f738d113f17 in operator new(unsigned long) (/lib/x86_64-linux-gnu/libasan.so.6+0xb1f17)
    #1 0x55d88c48053b in create_LL(std::vector<node*, std::allocator<node*> >&, int) (/home/osboxes/CSCE313/buggy-original
+0x253b)
    #2 0x55d88c48094c in main (/home/osboxes/CSCE313/buggy-original+0x294c)
    #3 0x7f738cca4cb1 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x28cb1)

SUMMARY: AddressSanitizer: 48 byte(s) leaked in 3 allocation(s).
```

After deleting the allocated dynamically memory, using the AddressSanitizer, run the program. It works well.

```
osboxes@osboxes:~/CSCE313$ g++ buggy-original.cpp -o buggy-original -fsanitize=address
osboxes@osboxes:~/CSCE313$ ./buggy-original
The sum of nodes in LL is 3
osboxes@osboxes:~/CSCE313$
```

10. [Using IDE]

The screenshot of debugging inside VS Code:



Final Code:

```
#include <iostream>
#include <vector>

using namespace std;
 //Blank A
class node {
public:
 //Blank B
        int val;
        node* next;
};

void create_LL(vector<node*>& mylist, int node_num){
   mylist.assign(node_num, NULL);

   //create a set of nodes
   for (int i = 0; i < node_num; i++) {
      //Blank C
      mylist[i] = new node();
      mylist[i]->val = i;
      mylist[i]->next = NULL;
   }
```

```
    //create a linked list
    for (int i = 0; i < node_num-1; i++) {
        mylist[i]->next = mylist[i+1];
    }
}

int sum_LL(node* ptr) {
    int ret = 0;
    while(ptr != NULL) {
        ret += ptr->val;
        ptr = ptr->next;
    }
    return ret;
}

int main(int argc, char ** argv){
    const int NODE_NUM = 3;
    vector<node*> mylist;

    create_LL(mylist, NODE_NUM);
    int ret = sum_LL(mylist[0]);
    cout << "The sum of nodes in LL is " << ret << endl;

    //Step4: delete nodes
    //Blank D
    for (int i = 0; i < mylist.size(); i++) {
            delete mylist[i];
    }
}
```