

PA0 Report by SIYUAN YANG

Coding Part:

```
1  #include <iostream>
2  // Blank A
3  #include <vector>
4  using namespace std;
5
6  class node {
7      // Blank B
8      public:
9          int val;
10         node* next;
11 };
12 void create_LL(vector<node*>& mylist, int node_num) {
13     mylist.assign(node_num, NULL);
14     //create a set of nodes
15     for (int i = 0; i < node_num; i++) {
16         //Blank C
17         mylist[i] = new node();
18         mylist[i]->val = i;
19         mylist[i]->next = NULL;
20     }
21     //create a linked list
22     for (int i = 0; i < node_num; i++) {
23         mylist[i]->next = mylist[i+1];
24     }
25 }
26 int sum_LL(node* ptr) {
27     int ret = 0;
28     while (ptr) {
29         ret += ptr->val;
30         ptr = ptr->next;
31     }
32     return ret;
33 }
34 int main(int argc, char ** argv){
35     const int NODE_NUM = 3;
36     vector<node*> mylist;
37     create_LL(mylist, NODE_NUM);
38     int ret = sum_LL(mylist[0]);
39     cout << "The sum of nodes in LL is " << ret << endl;
40
41     //Step4: delete nodes
42     //Blank D
43     for(int i = 0; i < NODE_NUM; i++)
44     {
45         delete mylist[i];
46         mylist[i] = NULL;
47     }
48     return 0;
49 }
```

Report:

Before we debug the code, we first need to fill in Blank A and B with “#include <vector>” and “public:” to make sure the required header files are included and allow variables to be used by outside of a class. After that, we start to debug the code by compiling with “g++ -g buggy.cpp” and go into the gdb mode by entering “gdb a.out”. We first run the whole program and find out that there are segmentation faults in the program and by entering “backtrace” can we find the last statement with the error which in Line 15. Then, I set a breakpoint at line 15 and again run the program inside gdb and figure out that we need to first initialize “mylist[i]” with new node and then dereference “mylist[i]” as a pointer to set the value and its next pointer to avoid dereferencing a null pointer. Similarly, we need to deal with sum_LL and create_LL by dereferencing mylist[i]. Finally, at blank D, by using AddressSanitizer can we tell that there is memory leak as we haven’t deallocated the memory from the heap for nodes of mylist. Thus, we use a for loop to delete all the nodes in mylist and set them to the null pointer to avoid dangling pointer. Overall, the program is working well.