COMP 4736 Lab 2  
Marco Ho Set T  
A01338160

1. The program appears to be crashing on line 13 as it’s trying to pass a null pointer to standard output.  
   A computer screen shot of a program

   Description automatically generated
2. The logical error would be due to the program updating the value of last prior to assigning it to second\_last. This makes it so both last and second\_last are assigned the value stored in ‘next’.  
   A computer screen with a red arrow pointing to the end of a black screen

   Description automatically generated
3. A screenshot of a computer program

   Description automatically generated

Errors:

1. Error: “Syscall param write(buf) points to uninitialized byte(s)”.  
   Explanation: The array being passed to write() in line 19 (memory\_bugs.c:19) has been declared by not defined; the array currently contains 10 nulls at the time of passing it to the write function and printing null(s) in C causes an error.
2. Error: “Invalid write of size 1”  
   Explanation: This error occurred at line 26 (memory\_bugs.c:26) because the program is trying to store character ‘A’ at the memory space where pointer *P* is pointing to despite after having ‘freed’ (released) the memory space on line 23.
3. Error: “Invalid read of size 1”  
   Explanation: Like error 2, on line 29 (memory\_bugs.c:29), the program is trying access memory space that has already been released.
4. Error: “Invalid free() / delete / delete[] / realloc()”  
   Explanation: On line 35 (memory\_bugs.c:35), the program is invoking free() on arr which is an int array of size 10 that was declared but never defined nor was it ever allocated with malloc/calloc/realloc.
5. Error: “30 bytes in 1 blocks are definitely lost in loss record 1 of 2”  
   Explanation: 30 bytes were allocated with malloc and assigned to *p* on line 16 (memory\_bugs.c:16) but then allocates 12 different bytes to the same variable *p* on line 22 (memory\_bugs.c:22) without invoking free() and releasing the 30 initial bytes.
6. Error: “50 bytes in 1 blocks are definitely lost in loss record 2 of 2”  
   Explanation: 50 bytes were allocated and the address of the 50 bytes were stored in pointer *q* in line 32 (memory\_bugs.c:32) but were never free’d/released.