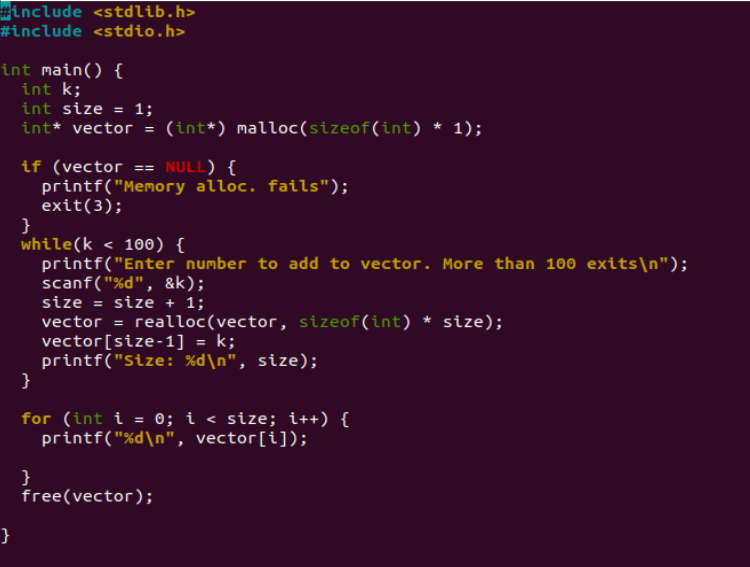
**Homework Wan Huzaifah bin Wan Azhar**

In this homework, you will gain some familiarity with memory allocation. First, you’ll write some buggy programs (fun!). Then, you’ll use some tools to help you find the bugs you inserted. Then, you will realize how awesome these tools are and use them in the future, thus making yourself more happy and productive. These tools are the debugger (e.g., gdb) and memory bug detector called valgrind.

**Answer:**

1. Segmentation fault happens when null run.
2. Gdb shows that SIGSEGV occurs at in main() function
3. Valgrind shows that the program is terminated with Segfault. This happens because there is an access at address 0x0 but it is not mapped. When null.c is declared with NULL, but not allocated any memory. The pointer points to address 0 (NULL). But address 0 is not declared or allocated in the program, so Segfault happens when dereferencing.
4. Run normally: nothing happens.  
   Run using gdb: no important output.  
   Run using valgrind: 4 bytes are lost. Memory leaks of 4 bytes occurs.
5. Segmentation fault happens when running normally.   
   There are two error when running the program through Valgrind:
   1. There are 400 bytes memory leaks, which 100 int in the program.
   2. Invalid write, which is data[100] = 0 when array data only allocated 0 to 99.
6. Segmentation fault happens when running normally.  
   When running through Valgrind, there are memory leaks like 5. And the process terminate because of invalid read. This invalid read is because the address are freed before trying to print the array.
7. Segmentation fault if run normally. Yes, definitely need tools to find out what type of error happens from segmentation fault. In this case, invalid free().
8. Vector-like structure using realloc:



This implementation works, but it can prone to error, compared to using dynamic array.