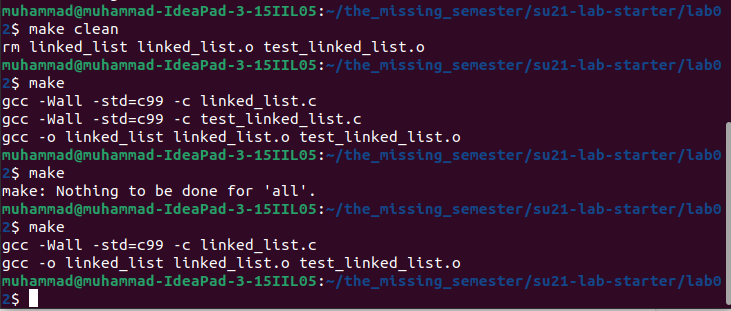
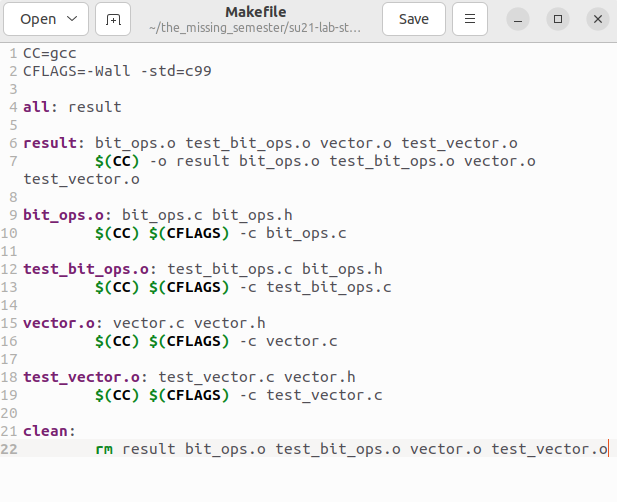
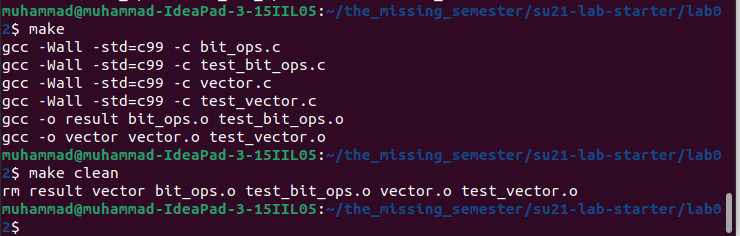
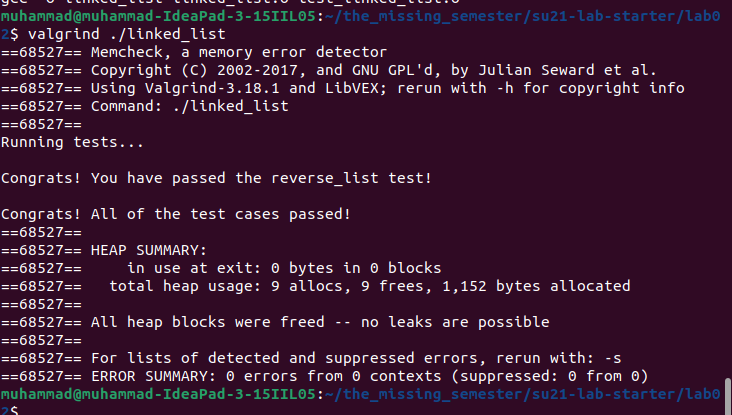
EXERCISE 1

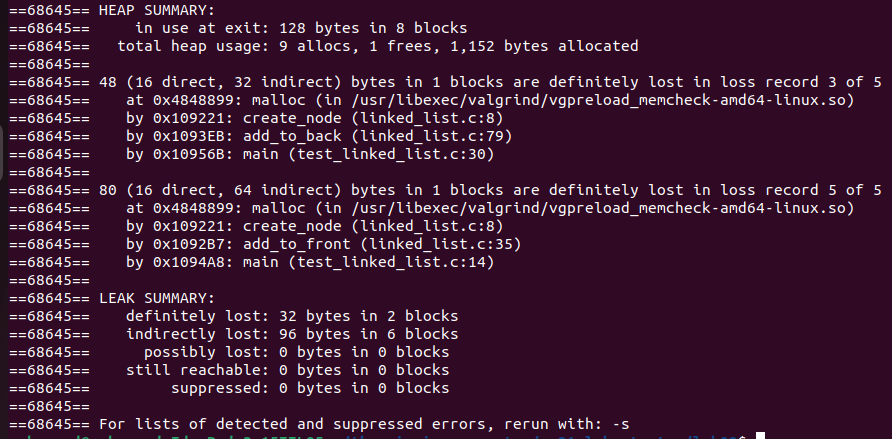




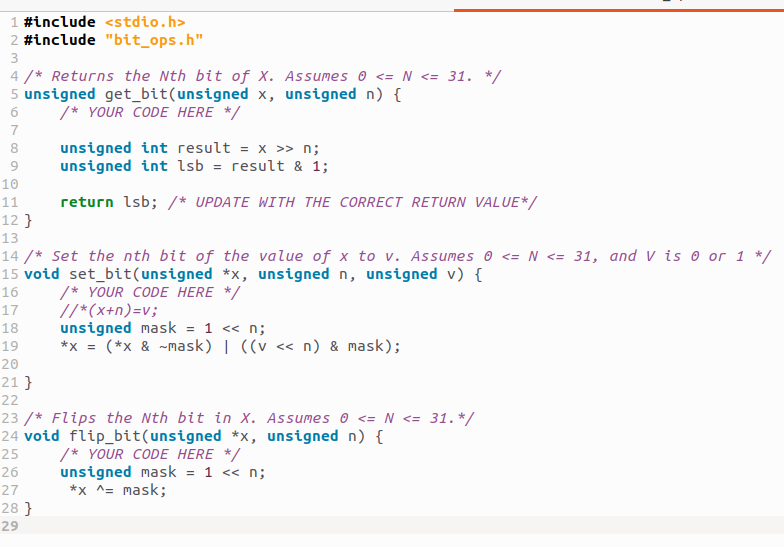


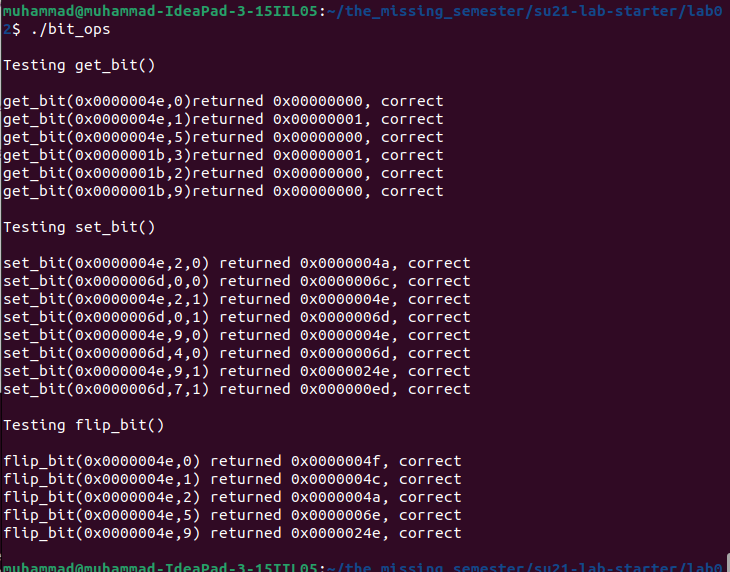
EXERCISE 2



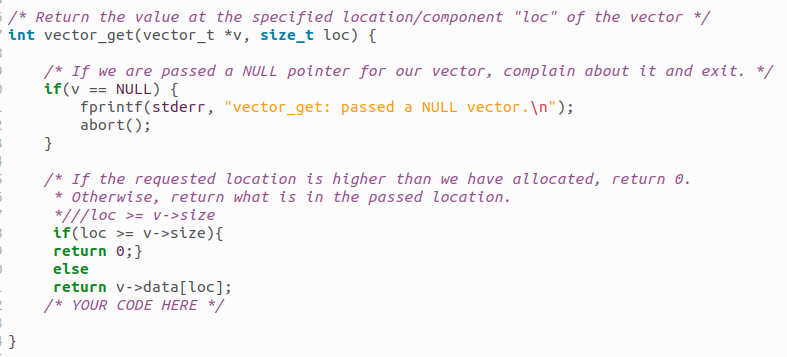
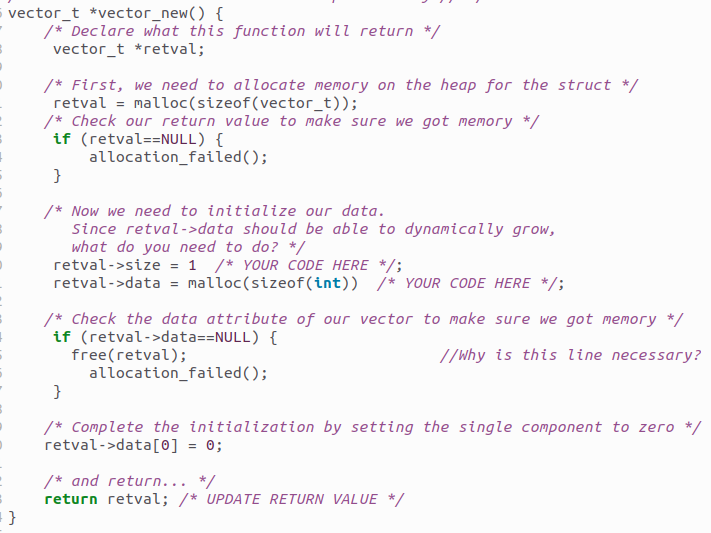


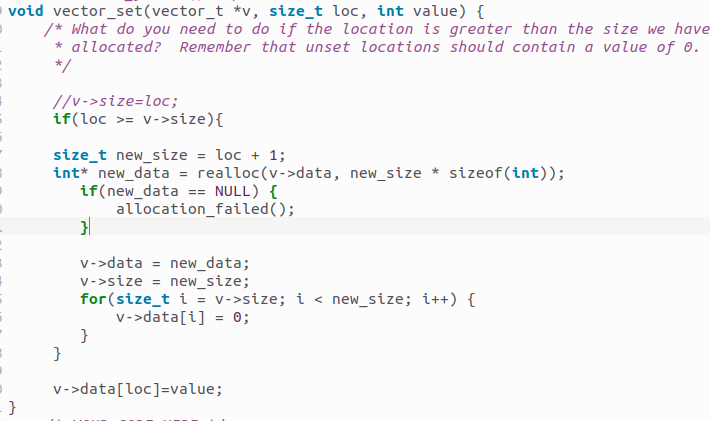
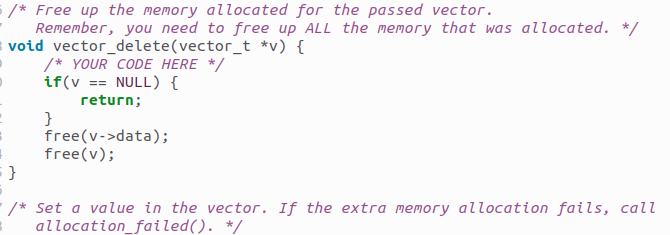
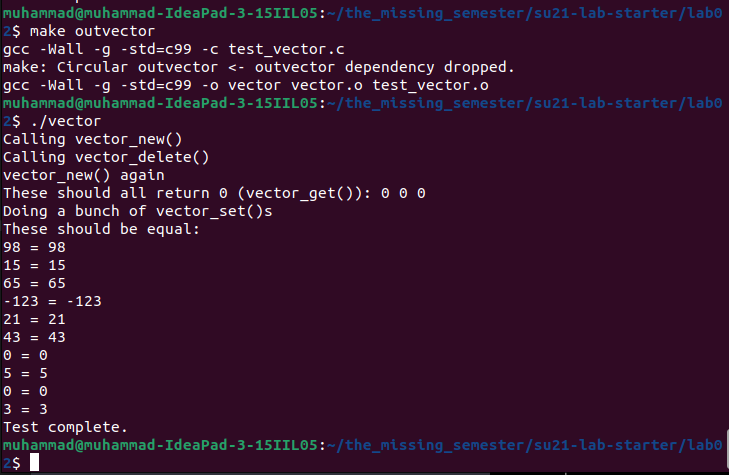
EXERCISE 3

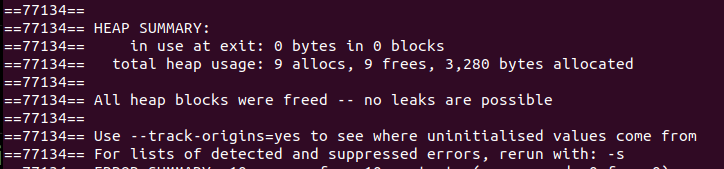




EXERCISE 4







vector\_t \*bad\_vector\_new() :

It is considered bad because it returns the pointer to a local variable stored in stack which will deallocate automatically when the function bad\_vector\_new() returns. This can lead to undefined behavior when the returned pointer is accessed outside the scope of the function, and result in crashes, data corruption, or other unexpected behavior in the program.

vector\_t also\_bad\_vector\_new():

It is considered bad because it return the value of struct instead of pointer. When a struct is returned by value, a copy of the struct is created on the stack, and the original struct inside the function is destroyed when the function returns. So, This can be inefficient in terms of memory usage and performance, especially if the struct contains a large amount of data.

Git Hub Repo Link: