Workshop 3

Members –

Name : Zwe Nyan Win

ID : 6540179

Name : Win Yu Maung

ID: 6612054

Name : Zin Me Me That

ID : 6611706

## Task 1:

1. Where each variable is stored 🡪

* counter is stored in static storage (global variable, stored in data segment)
* x is stored in stack (local variable inside foo(),it is allocated when function is called)
* p is stored in stack (the pointer itself which is local to foo()).
* The integer pointed to by p(\*p) is stored in heap.

1. When is the memory block allocated and deallocated for each variable? 🡪

* Counter is allocated during program compile time and is deallocated in program termination time.
* x and p is allocated during runtime when function is called and is deallocated when the function call is ended.
* \*p is allocated when malloc() is called , and is deallocated when free(p) is called manually.

1. If free(p) is not called before foo() ends, the memory allocated to \*p will not be cleared which will result in memory leaks and program will become heavy as it runs because C doesn’t have GC to clean unused variables.

## 

## Task 3

1. Why do local variable disappear after a function end 🡪

* Since local variable are stored in C default storage such as stacks, it will be disappeared after a function ends.

1. How is memory management in Python or ML different from C? 🡪

* **C:** Manual memory management is allocated with malloc() and disallocated with free(). There is no garbage collector, we have to manage memory manually.If free isn’t called after malloc() is used, memory leaks can happen.
* **Python/ML: There is**  an automatic garbage collection which frees unused objects when it is no longer referenced automatically, but leaks can still happen if references persist.