Q.What are defer, panic and recover? How to recover from a runtime panic?

A **defer statement** pushes a function call onto a list. The list of saved calls is executed after the surrounding function returns. Defer is commonly used to simplify functions that perform various clean-up actions. For example, defer statements guarantees that regardless of the number of return statements in the function, the files which were opened are closed. Deferred function calls are executed in Last in First Out order after the surrounding function returns.

**Panic** is a built-in function that stops the ordinary flow of control of a program and starts panicking. Whenever a function say f() calls panic the execution of function f() stops, if there are any deferred functions in f() they are executed normally and the f() returns to its caller. To the caller, f() behaves like a call to panic function and then the program stack begins unwinding at a point where the program crashes. Panics can be initiated directly or can also be caused by runtime errors like out of bound array accesses.

The **recover function** stops the chain reaction of panic from going farther up the call stack. The recover function can only be used inside of a defer function. Because during the panic chain reaction only defer functions will be executed. If the recover function is called and there is no panic occurring, the recover function will return nil. If there is a panic occurring, then the panic is stopped and the value given to the panic call will be returned.

If a runtime panic occurs, it can be handled with the **help of a recover function which is written inside a defer function** because defer function executes even after panic occurs and hence writing a recover function under defer would be able to handle runtime panic.

.