Slip 1 Q1. A) Write a program in GO language to accept user choice and print answers using arithmetic operators.

```
package main
import "fmt"
func main(){
var a,b,choice int

fmt.Println("Enter two Numbers")
fmt.Scan(&a,&b)

fmt.Println("Choice an operation")
fmt.Println("1 :Add")
fmt.Println("2 :Sub")
fmt.Println("3 :Mul")
fmt.Println("4 :Div")
fmt.Println("Enter your choice")
fmt.Scan(&choice)

switch choice{
case 1:
fmt.Println("Result",a+b)
case 2:
fmt.Println("Result",a-b)
case 3:
fmt.Println("Result",a*b)
case 4:
fmt.Println("Result",a/b)
default:
fmt.Println("Invalid choice")
}
```

```
}
```

Slip 2 Q1. A) Write a program in GO language to print Fibonacci series of n terms.

```
package main
import "fmt"
func main(){
var n,a,b,next int
fmt.Print("Enter number of terms")
fmt.Scan(&n)

a,b= 0,1
fmt.Println("Fibinacci Series")
for i:=0 ;i<n;i++{
fmt.Print(a,"")
next=a+b
a=b
b=next
}
}
```

Slip 3 Q1. A) Write a program in the GO language using function to check whether accepts number is palindrome or not

```
package main
import "fmt"

func isPalindrome(num int)(int){
var rem,rev int
n:=num
```

```
for n>0{
rem=n%10
rev=rev*10+rem
n=n/10
}
return rev
}

func main(){
fmt.Println("Enter no")
var num int
fmt.Scan(&num)

rev:=isPalindrome(num)
if(rev==num){
fmt.Println("Yes")
}else{
fmt.Println("No")
}
}
```

Slip 4 Q1. A) Write a program in GO language to print a recursive sum of digits of a given number

```
package main

import "fmt"

func main(){
var num ,sum int
fmt.Println("Enter no")
```

```
fmt.Scan(&num)

for num>0{
sum =sum+num%10
num =num/10
}
fmt.Println(sum)
}
```

Slip 5 Q1. A) Write a program in GO language program to create Text file

```
package main

import "os"

func main() {
        os.Create("king.txt")
}
```

Slip 6 Q1. B) Write a program in GO language to copy all elements of one array into another using a method

```
package main

import "fmt"

func copyArray(src []int) []int {
dest := make([]int, len(src))
copy(dest, src)
return dest
}
```

```
func main() {
original := []int{1, 2, 3, 4, 5}
copied := copyArray(original)

fmt.Println("Original Array:", original)
fmt.Println("Copied Array:", copied)
}
```

Slip 7 Q1. B) Write a program in GO language to create structure student. Writea method show() whose receiver is a pointer of struct student

```
package main
import "fmt"

type Student struct {
rollno int
name string
marks int
}

func (s *Student) show() {
fmt.Printf("Roll no:%d, Name:%s,Marks:%d",s.rollno,s.name,s.marks)
}

func main(){
stud := Student{
        rollno: 101,
        name: "rohit",
         marks: 85,
         }
```

```
stud.show()
}
```

Slip 8 Q1. A) Write a program in GO language to accept the book details such as BookID, Title, Author, Price. Read and display the details of 'n' number of books

```
package main

import "fmt"

type Book struct{
BookId int
title string
price int
}

func main() {
var n int
fmt.Print("Enter the number of books")
fmt.Scan(&n)

books := make([]Book, n)

for i := 0; i < n; i++ {
fmt.Print("Enter details of books")
fmt.Scan(&books[i].BookId, &books[i].title, &books[i].price)
}

fmt.Println("Books details")
for _, book := range books {
fmt.Printf("BookId: %d, Title:%s, price:%d",book.BookId, book.title, book.price)
```

```
}
}
```

Slip 9 Q1. A) Write a program in GO language using a function to check whether the accepted number is palindrome or not

```
package main
import "fmt"

func isPalindrome(num int)(int){
var rem,rev int
n:=num

for n>0{
rem=n%10
rev=rev*10+rem
n=n/10
}
return rev
}

func main(){
fmt.Println("Enter no")
var num int
fmt.Scan(&num)

rev:=isPalindrome(num)
if(rev==num){
fmt.Println("Yes")
}else{
fmt.Println("No")
```

```
}
}
```

Slip 10 Q1. A) Write a program in GO language to create an interface and display its values with the help of type assertion.

```
package main

import "fmt"

func main() {
   checkType("Hello, Go!")
   checkType(100)
   checkType(3.14)
}

func checkType(i interface{}) {
   if v, ok := i.(string); ok {
      fmt.Println("String:", v)
   } else {
      fmt.Println("Not a string:", i)
   }
}
```

Slip 11 Q1. A) Write a program in GO language to check whether the accepted number is two digit or not

```
package main
import "fmt"

func main() {
var num int
```

```
fmt.Print("Enter a number ")
fmt.Scan(&num)

if num >= 10 && num <= 99 || num >= -10 && num <= -99 {

fmt.Println("It is two digit number")
}else{
fmt.Println("It is not two digit number")
}
}
```

Slip 12 Q1. A) Write a program in GO language to swap two numbers using call by reference concept

```
package main

import "fmt"

func swap(a,b *int){
*a, *b = *b , *a
}

func main(){
var x,y int
fmt.Println("Enter two digits")
fmt.Scan(&x,&y)

fmt.Println("Before swaping x=",x,"y=",y)
swap(&x,&y)
fmt.Println("After swaping x=",x,"y=",y)
}
```

Slip 13 Q1. A) Write a program in GO language to print sum of all even and odd numbers separately between 1 to 100

```
package main
import "fmt"
func main(){
even , odd := 0, 0

for i:=1; i<=100; i++{
if i%2==0{
even+=i
}else{
odd+=i
}
}

fmt.Println(" sum of even number ",even)
fmt.Println("sum of odd number  ",odd)
}
```

Slip 14 Q1. A) Write a program in GO language to demonstrate working of slices (like append, remove, copy etc.)

```
package main
import "fmt"

func main(){
s:= []int{1,2,3}
s = append(s,4)
fmt.Println("Slice",s)
```

```
s = s[1:]
fmt.Println("After remove ",s)

c:= make([]int, len(s))
copy(c,s)
fmt.Println("Copied",c)
}
```

Slip 15 Q1. A) Write a program in GO language to demonstrate function return multiple values.

```
package main
import "fmt"

func myfunc(n1 int,n2 int)(int ,int){
sum:= n1+n2
sub:= n1-n2
return sum,sub
}

func main() {
Sum,Sub:=myfunc(20,10)
fmt.Printf("Sum %d,Sub %d",Sum,Sub)
}
```

Slip 16 Q1. B) Write a program in GO language that prints out the numbers from 0 to 10, waiting between 0 and 250 ms after each one using the delay function

```
**package main
```

```
import ("fmt"
"math/rand"
"time"
)

func main(){
for i:=0; i<=10 ;i++{
fmt.Println(i)
time.Sleep(time.Duration(rand.Intn(250))*time.Millisecond)

}
}**
```

Slip 17 Q1. A) Write a program in GO language to illustrate the concept of returning multiple values from a function. (Add, Subtract, Multiply, Divide)

```
package main
import (
"fmt"
)

func myFunc(a int,b int)(int,int,int,int){
sum:=a+b
sub:=a-b
mul:=a*b
div:=a/b
return sum,sub,mul,div
}

func main(){
Sum,Sub,Mul,Div:=myFunc(10,10)
```

```
fmt.Println(Sum,Sub,Mul,Div)
}
```

Slip 18 Q1. A) Write a program in GO language to print a multiplication table of number using function.

```
package main
import "fmt"

func printTable(num int){
for i:=1;i<=10;i++{
fmt.Printf("%d\\n",num*i)
}
}

func main(){
var num int
fmt.Print("Enter nnumber")
fmt.Scan(&num)
printTable(num)
}
```

Slip 19 Q1. A) Write a program in GO language to illustrate the function returning multiple values(add, subtract)

```
package main
import (
"fmt"
)

func myFunc(a int,b int)(int ,int){
add:=a+b
sub:=a-b
```

```
return add,sub
}

func main(){
Add,Sub:=myFunc(20,10)
fmt.Println(Add,Sub)
}
```

Slip 20 Q1. A) Write a program in Go language to add or append content at the end of a text file

```
package main

import (
	"os"
)

func main() {
	f,_:=os.OpenFile("example.txt",os.O_APPEND|os.O_CREATE|os.O_WRONLY, 0644)
	defer f.Close()
	f.Write([]byte("how"))
	}
```

IOT Q2 for all.

```
int lpin=13;
int dtime=10000;
void setup(){
// Put your setup code here,to run once:
pinMode(lpin,OUTPUT);
}
```

```
void loop(){
// Put your main code here, to run repeatedly:
digitalWrite(lpin,HIGH);
delay(dtime);
digitalWrite(lpin,LOW);
delay(dtime);
```