**INDEX**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **PROGRAMS** | **PAGE NO** |
| 1 | GO Program to Check Whether a Number is Even or Odd. | 1-2 |
| 2 | GO program to display of standard arithmetic operators with 2 integer values. | 3-4 |
| 3 | GO Program to Find LCM and GCD of given two numbers. | 5-7 |
| 4 | GO Program to find the index of first occurrence of a substring. | 8-9 |
| 5 | GO Program to get first and last element of slice in Golang. | 10-11 |
| 6 | GO Program to get total number of characters in a string. | 12-13 |
| 7 | GO Program to print full Pyramid using STAR | 14-15 |
| 8 | GO program for implementation of Binary Search. | 16-17 |
| 9 | GO program for implementation of Linear Search. | 18-19 |
| 10 | GO Program to Generate Multiplication Table | 20-21 |
| 11 | GO Program to Add Two Matrix Using Multi-dimensional Arrays. | 22-24 |
| 12 | GO Program to Calculate Area of Rectangle and Square. | 25-26 |
| 13 | GO Program to Check Whether a Number is Palindrome or Not. | 27-28 |
| 14 | GO program to implementation of Tower of Hanoi Algorithm. | 29-30 |
| 15 | GO Program to print the ascii code for each letter in the alphabet. | 31-32 |
| 16 | GO Program to read file line by line to string. | 33-34 |
| 17 | GO Program to take user input and addition of two strings | 35-36 |
| 18 | GO Program to Get current date and time in various format in golang. | 37-38 |
| 19 | GO program with example of Array Reverse Sort Functions for integer and strings | 39-40 |
| 20 | GO Program to replace substrings in a string. | 41-42 |

**1. GO Program to Check Whether a Number is Even or Odd.**

package main

import "fmt"

func main() {

fmt.Println("Enter number:")

var n int

fmt.Scanln(&n)

if n%2==0 {

fmt.Println(n,"is Even number")

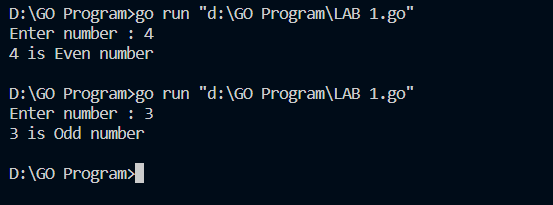
}else{

fmt.Println(n,"is odd number")

}

}

**Output: -**



**2. GO program to display of standard arithmetic operators with 2 integer values.**

package main

import "fmt"

func main() {

fmt.Println("1 + 2 = ", 1+2)

fmt.Println("5 \* 2 = ", 5\*2)

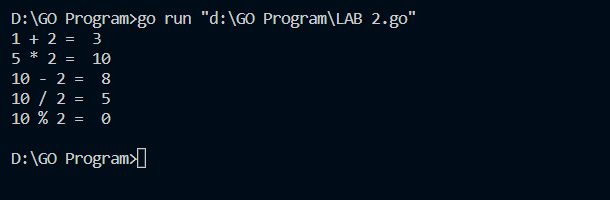
fmt.Println("10 - 2 = ", 10-2)

fmt.Println("10 / 2 = ", 10/2)

fmt.Println("10 % 2 = ", 10%2)

}

**Output: -**



**3. GO Program to Find LCM and GCD of given two numbers.**

package main

import "fmt"

func lcm(temp1 int,temp2 int){

var lcmnum int=1

if temp1>temp2 {

lcmnum=temp1

} else {

lcmnum=temp2

}

for {

if lcmnum%temp1==0&& lcmnum%temp2==0 {

fmt.Printf("LCM of %d and %d is %d",temp1,temp2,lcmnum)

break

}

lcmnum++

}

return

}

func gcd(temp1 int,temp2 int) {

var gcdnum int

for i:=1;i<=temp1 && i<=temp2;i++ {

if temp1%i==0 && temp2%i==0 {

gcdnum=i

}

}

fmt.Printf("GCD of %d and %d is %d",temp1,temp2,gcdnum)

return

}

func main() {

var n1,n2,action int

fmt.Println("Enter two positive integers:")

fmt.Scanln(&n1)

fmt.Scanln(&n2)

fmt.Println("Enter 1 for LCM and 2 for GCD")

fmt.Scanln(&action)

switch action {

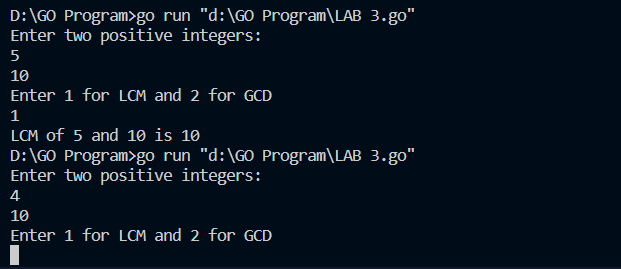
case 1:lcm(n1,n2)

case 2:gcd(n1,n2)

}

}

**Output: -**



**4. GO Program to find the index of first occurrence of a substring.**

package main

import (

"fmt"

"strings"

)

func main() {

str := "the cat in the hat"

i := strings.Index(str, "hat")

fmt.Println(i)

var str2 string

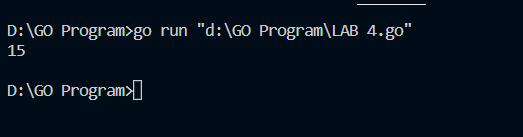
str2="The cat in the hat"

j:=strings.Index(str2,"in")

fmt.Println(j)

}

**Output: -**



**5. GO Program to get first and last element of slice in Golang.**

package main

import "fmt"

func main() {

intSlice := []int{1, 2, 3, 4, 5}

fmt.Println("slice: ", intSlice)

last := intSlice[len(intSlice)-1]

fmt.Println("last element: ", last)

first := intSlice[:1]

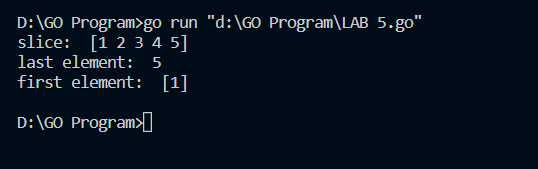
fmt.Println("first element: ", first)

remove := intSlice[:len(intSlice)-1]

fmt.Println("remove last: ", remove)

}

**Output: -**



**6. GO Program to get total number of characters in a string.**

package main

import "fmt"

func main(){

str:="Adithya GB"

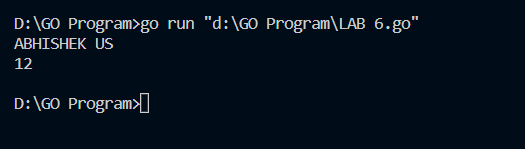
fmt.Println(str)

len:=len(str)

fmt.Println(len)

}

**Output: -**



**7. GO Program to print full Pyramid using STAR.**

package main

import "fmt"

func main() {

var rows int

var k int=0

fmt.Print("Enter number of rows:")

fmt.Scanln(&rows)

for i:=1;i<=rows;i++ {

k=0

for space:=1;space<=rows-i;space++ {

fmt.Print(" ")

}

for {

fmt.Print("\*")

k++

if k==2\*i-1{

break

}

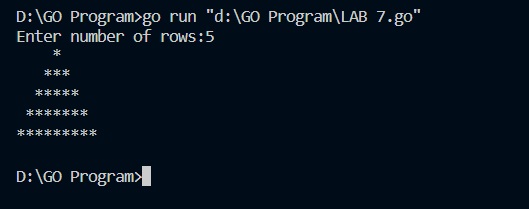
}

fmt.Println("")

}

}

**Output: -**

****

**8. GO program for implementation of Binary Search.**

package main

import "fmt"

func binarySearch(needle int,haystack []int)bool {

low:=0

high:=len(haystack)

for low<=high {

median:=(low+high)/2

if haystack[median]<needle {

low=median+1

} else {

high=median-1

}

}

if low==len(haystack)||haystack[low]!=needle {

return false

}

return true

}

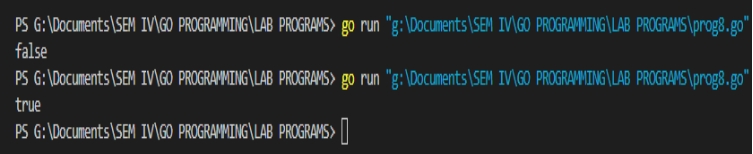
func main() {

items:=[]int{1,2,9,20,31,45,63,70,100}

fmt.Println(binarySearch(100,items))

}

**Output: -**



**9. GO program for implementation of Linear Search.**

package main

import "fmt"

func linearSearch(datalist []int,key int)bool {

for \_,item:=range datalist {

if item==key {

return true

}

}

return false

}

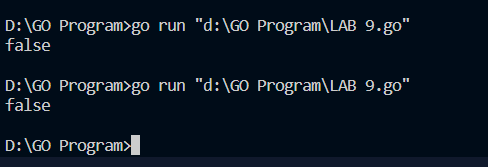
func main() {

items:=[]int{95,78,56,84,25,35,15,26}

fmt.Println(linearSearch(items,96))

}

**Output: -**

****

**10. GO Program to Generate Multiplication Table.**

package main

import "fmt"

func main() {

var n int

fmt.Print("enter the integer number:")

fmt.Scanln(&n)

i:=1

for {

if i>10 {

break

}

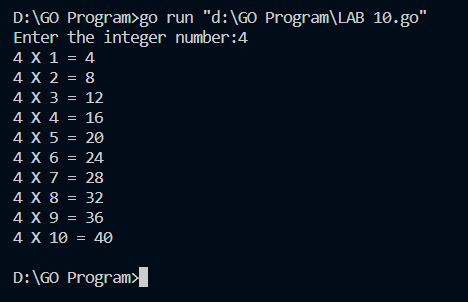
fmt.Println(n,"X",i,"=",n\*i)

i++

}

}

**Output: -**

****

**11. GO Program to Add Two Matrix Using Multi-dimensional Arrays**.

package main

import "fmt"

func main() {

var matrix1 [100][100]int

var matrix2 [100][100]int

var sum [100][100]int

var row,col int

fmt.Println("Enter number of rows:")

fmt.Scanln(&row)

fmt.Println("Enter number of cols:")

fmt.Scanln(&col)

fmt.Println()

fmt.Println("==========matrix1==========")

fmt.Println()

for i:=0;i<row;i++ {

for j:=0;j<col;j++ {

fmt.Printf("Enter the element for matrix1 %d%d:",i+1,j+1)

fmt.Scanln(&matrix1[i][j])

}

}

fmt.Println()

fmt.Println("==========matrix2==========")

fmt.Println()

for i:=0;i<row;i++ {

for j:=0;j<col;j++ {

fmt.Println("Enter the element for matrix2 %d%d:",i+1,j+1)

fmt.Scanln(&matrix2[i][j])

}

}

for i:=0;i<row;i++ {

for j:=0;j<col;j++ {

sum[i][j]=matrix1[i][j]+matrix2[i][j]

}

}

fmt.Println()

fmt.Println("==========Sum of Marix==========")

fmt.Println()

for i:=0;i<row;i++ {

for j:=0;j<col;j++ {

fmt.Printf("%d",sum[i][j])

if j==col-1 {

fmt.Println("")

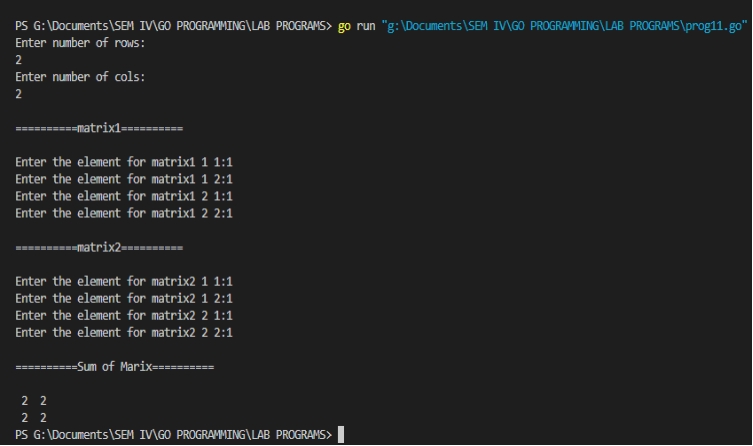
}

}

}

}

**Output: -**



**12. GO Program to Calculate Area of Rectangle and Square.**

package main

import "fmt"

var area int

func main() {

var l,b int

fmt.Println("Enter length of rectangle:")

fmt.Scanln(&l)

fmt.Println("Enter the bredth of rectangle:")

fmt.Scanln(&b)

area=l\*b

fmt.Println("Area of rectangle:",area)

fmt.Println("Enter length of square:")

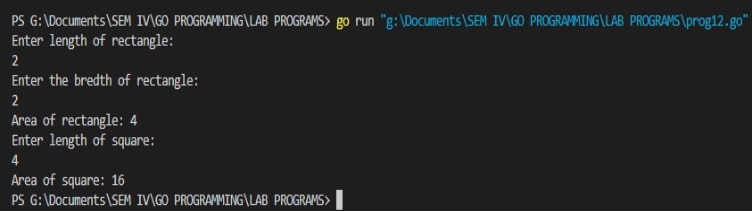
fmt.Scanln(&l)

area=l\*l

fmt.Println("Area of square:",area)

}

**Output: -**



**13. GO Program to Check Whether a Number is Palindrome or Not.**

package main

import "fmt"

func main() {

var number,remainder,temp int

var reverse int=0

fmt.Println("Enter any positive integer:")

fmt.Scanln(&number)

temp=number

for {

remainder=number%10

reverse=reverse\*10+remainder

number/=10

if number==0 {

break

}

}

if temp==reverse {

fmt.Printf("%d is a palindrome",temp)

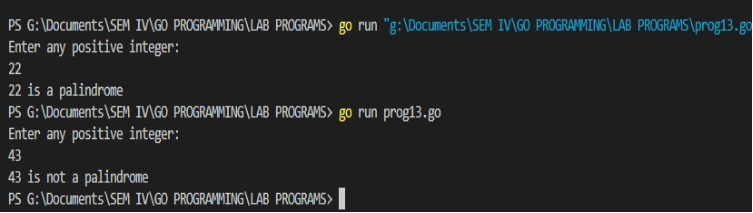
} else {

fmt.Printf("%d is not a palindrome",temp)

}

}

**Output: -**



**14. GO program to implementation of Tower of Hanoi Algorithm.**

package main

import "fmt"

type solver interface {

play(int)

}

type towers struct {

}

func(t\*towers)play(n int) {

t.moveN(n,1,2,3)

}

func(t\*towers)moveN(n,from,to,via int){

if n>0 {

t.moveN(n-1,from,via,to)

t.moveM(from,to)

t.moveN(n-1,via,to,from)

}}

func(t\*towers)moveM(from,to int) {

fmt.Println("move disk from rod",from,"to rod",to)}

func main() {

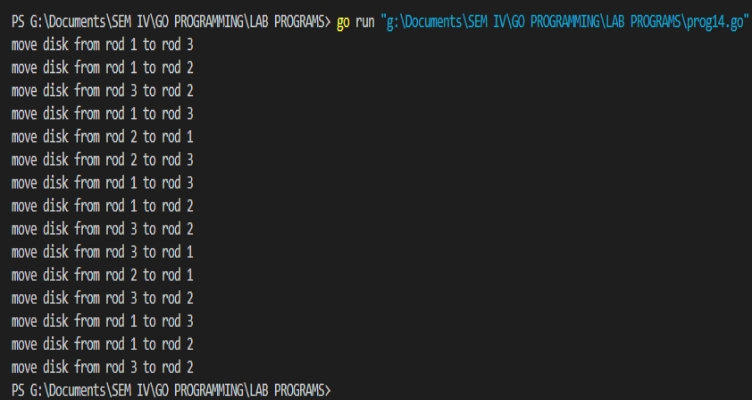
var t solver

t=new(towers)

t.play(4)

}

**Output: -**



**15. GO Program to print the ascii code for each letter in the alphabet.**

package main

import "fmt"

func main() {

var str="abcdefghijklmnopqrstuvwxyz"

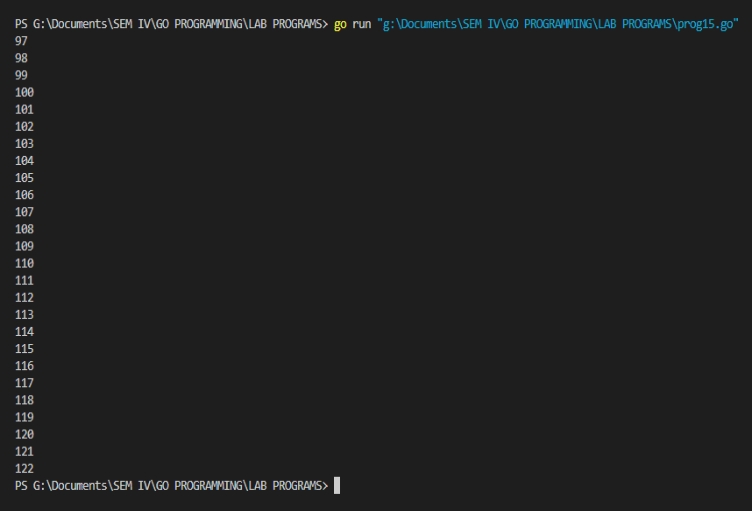
for \_,c:=range str {

fmt.Println(c)

}

}

**Output: -**



**16. GO Program to read file line by line to string.**

package main

import (

"bufio"

"fmt"

"log"

"os"

)

func main() {

file,err:=os.Open("kotlin.txt")

if err!=nil {

log.Fatalf("failed opening file:%s",err)

}

scanner:=bufio.NewScanner(file)

scanner.Split(bufio.ScanLines)

var txtlines []string

for scanner.Scan() {

txtlines=append(txtlines,scanner.Text())

}

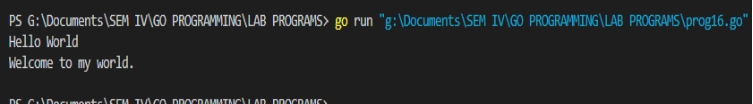
file.Close()

for \_, eachline := range txtlines {

fmt.Println(eachline)

}

**Output: -**



**17. GO Program to take user input and addition of two strings.**

package main

import "fmt"

func main() {

fmt.Println("Enter first string:")

var first string

fmt.Scanln(&first)

fmt.Println("Enter second string:")

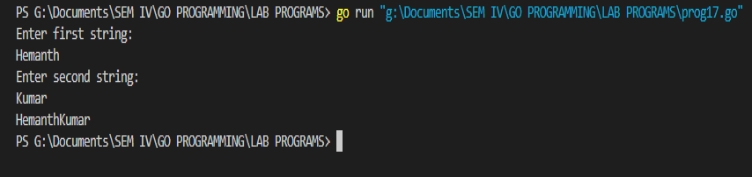
var second string

fmt.Scanln(&second)

fmt.Println(first+second)

}

**Output: -**



**18.GO Program to Get current date and time in various format in golang.**

package main

import (

"fmt"

"time"

)

func main() {

currentTime:=time.Now()

fmt.Println("Current Time in String:",currentTime.String())

fmt.Println("MM-DD-YYYY:",currentTime.Format("01-02-2006"))

fmt.Println("YYYY-MM-DD:",currentTime.Format("2006-01-02"))

fmt.Println("YYYY.MM.DD:",currentTime.Format("2006.01.02 15:04:05"))

fmt.Println("YYYY#MM#DD {Special Character}:",currentTime.Format("2006#01#02"))

fmt.Println("YYYY-MM-DD hh:mm:ss :",currentTime.Format("2006-01-02 15:04:05"))

fmt.Println("Time with MicroSeconds:",currentTime.Format("2006-01-02 15:04:05.000000"))

fmt.Println("Time with NanoSeconds:",currentTime.Format("2006-01-02 15:04:05.000000000"))

fmt.Println("ShortNum Month:",currentTime.Format("2006-1-02"))

fmt.Println("LongMonth:",currentTime.Format("2006-January-02"))

fmt.Println("ShortMonth:",currentTime.Format("2006-Jan-02"))

fmt.Println("Short Year:",currentTime.Format("06-Jan-02"))

fmt.Println("LongWeekDay:",currentTime.Format("2006-01-02 15:04:05 Monday"))

fmt.Println("ShortWeekDay:",currentTime.Format("2006-01-02 Mon"))

fmt.Println("ShortDay:",currentTime.Format("Mon 2006-01-2"))

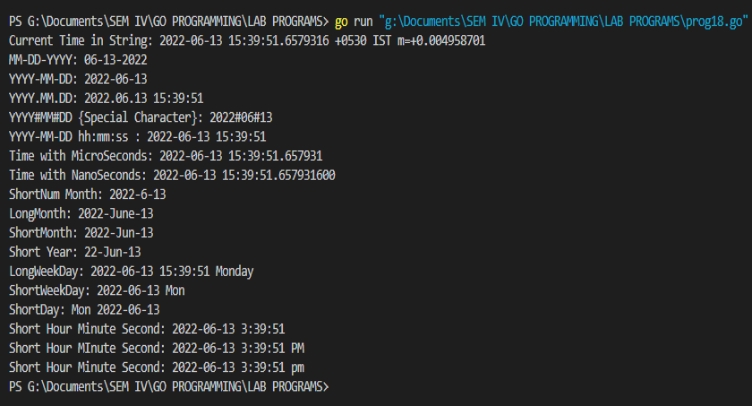
fmt.Println("Short Hour Minute Second:",currentTime.Format("2006-01-02 3:4:5"))

fmt.Println("Short Hour MInute Second:",currentTime.Format("2006-01-02 3:4:5 PM"))

fmt.Println("Short Hour Minute Second:",currentTime.Format("2006-01-02 3:4:5 pm"))

}

**Output: -**



**19.GO program with example of Array Reverse Sort Functions for integer and strings.**

package main

import (

"fmt"

"sort"

)

func main() {

fmt.Println("Integer reverse sort")

num:=[]int{50,40,60,9,80}

sort.Sort(sort.Reverse(sort.IntSlice(num)))

fmt.Println(num)

fmt.Println()

fmt.Println("Strings reverse sort")

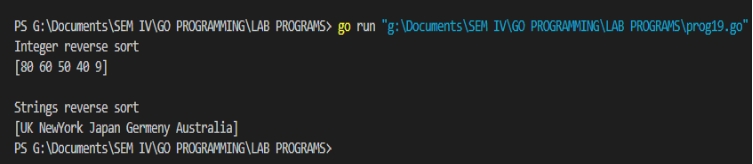
text:=[]string{"Japan","UK","Germeny","Australia","NewYork"}

sort.Sort(sort.Reverse(sort.StringSlice(text)))

fmt.Println(text)

}

**Output: -**



**20. GO Program to replace substrings in a string.**

package main

import (

"fmt"

"strings"

)

func main() {

str1:="A cat is a cat is a cat is a cat"

fmt.Println(str1)

str2:=strings.Replace(str1,"cat","dog",1)

fmt.Println(str2)

str3:=strings.Replace(str1,"cat","dog",2)

fmt.Println(str3)

str4:=strings.Replace(str1,"cat","dog",-1)

fmt.Println(str4)

}

**Output: -**

