**PROJECT PART 1**

**LEARNING AND EXPERIMENTATION**

**GO LANGUAGE**

1. **Program to find the nth Fibonacci Number.**

**Program: (fibonacci.go)**

package main

import "fmt"

func main() {

var n int;

var nthFib int;

fmt.Printf("Please enter a number: ")

fmt.Scanf("%d",&n)

nthFib= fibonacci(n)

fmt.Printf("The nth fibonacci number is : %d", nthFib);

}

func fibonacci(n int) int{

var fib int;

var n1 int=1;

var n2 int=1;

if(n==1 || n==2){

return 1;

}

for i:=3;i<=n;i++ {

fib = n1+n2;

n1= n2;

n2= fib;

}

return fib

}

**OUTPUT:**

**Run the following command in Command Prompt:** go run fibonacci.go

Please enter a number: 4

The nth fibonacci number is : 3

2. **Program to find the factorial of a number using recursion:**

**Program: (factorial\_recursion.go)**

package main

import "fmt"

func main() {

var n int;

var f int;

fmt.Printf("Enter a number:");

fmt.Scanf("%d",&n);

f = factorial(n)

fmt.Printf("The factorial of the number %d is : %d",n,f)

}

func factorial(n int) int{

if n==0{

return 1

}

return (n\* factorial(n-1))

}

**OUTPUT:**

**Run the following command in Command Prompt:** go run factorial\_recursion.go

Enter a number:5

The factorial of the number 5 is : 120

**3. Program to print Grade based on the given marks from Command Prompt: (grade.go)**

package main

import "os"

import "fmt"

import "strconv"

func main() {

   //Fetching 1st argument from the command line

   arg := os.Args[1]

   fmt.Println(arg)

   i, err := strconv.Atoi(arg)

   if err != nil {

       // handle error

       fmt.Println(err)

       os.Exit(2)

   }

   fmt.Println("Grade based on your marks is")

   switch {

   case  i < 60 && i >= 0:

    fmt.Println("E");

case i < 70:

    fmt.Println("D");

   case i < 80:

    fmt.Println("C");

   case i < 90:

    fmt.Println("B");

   case i <= 100:

    fmt.Println("A");

   default:

    fmt.Println("Invalid Input");

   }

}

**OUTPUT:**

**Run the following command in Command Prompt:** go run grade.go 92

92

Grade based on your marks is:

A

**4. Quick Sort Implementation which randomly takes few numbers using the rand package and sorts it in ascending order:**

**PROGRAM: (quicksort.go)**

package main

import (

"fmt"

"math/rand"

"time"

)

func qsort\_pass(arr []int, done chan int) []int{

if len(arr) < 2 {

done <- len(arr)

return arr

}

pivot := arr[0]

i, j := 1, len(arr)-1

for i != j {

for arr[i] < pivot && i!=j{

i++

}

for arr[j] >= pivot && i!=j{

j--

}

if arr[i] > arr[j] {

arr[i], arr[j] = arr[j], arr[i]

}

}

if arr[j] >= pivot {

j--

}

arr[0], arr[j] = arr[j], arr[0]

done <- 1;

go qsort\_pass(arr[:j], done)

go qsort\_pass(arr[j+1:], done)

return arr

}

func qsort(arr []int) []int {

done := make(chan int)

defer func() {

close(done)

}()

go qsort\_pass(arr[:], done)

rslt := len(arr)

for rslt > 0 {

rslt -= <-done;

}

return arr

}

func main() {

fmt.Println("About to sort.")

rand.Seed(time.Now().UTC().UnixNano())

arr\_rand := make([]int, 20)

for i := range arr\_rand {

arr\_rand[i] = rand.Intn(10)

}

fmt.Println(arr\_rand)

qsort(arr\_rand)

fmt.Println(arr\_rand)

}

**OUTPUT:**

**Run the following command in Command Prompt:** go run quicksort.go

About to sort.

[7 5 6 7 8 3 6 6 2 5 8 1 9 9 3 4 0 3 8 7]

[0 1 2 3 3 3 4 5 5 6 6 6 7 7 7 8 8 8 9 9]

**Program 5:**

**Program demonstrating concurrency and channel concepts (concurrency.go)**

package main

import (

"fmt"

"time"

)

func pinger(c chan string) {

for i := 0; ; i++ {

c <- "ping"

}

}

func printer(c chan string) {

for {

msg := <- c

fmt.Println(msg)

time.Sleep(time.Second \* 1)

}

}

func main() {

var c chan string = make(chan string)

go pinger(c)

go printer(c)

go ponger(c)

var input string

fmt.Scanln(&input)

}

func ponger(c chan string) {

for i := 0; ; i++ {

c <- "pong"

}

}

**OUTPUT:**

**Execute the following command in command prompt:** go run concurrency.go

ping

pong

ping

pong

**Please break by pressing the CTRL + C or press enter.**