**Golang Skills:**

1. Rest API standards and how they have developed Microservices using Golang (middleware, service to service communication, HTTP status codes, Different HTTP methods and it’s uses and sample program to create POST and GET API.)
2. Concepts from Golang (Goroutine, Channel, Wait Group, Select statement, Slice, Map, interfaces etc.)
3. Program based on Goroutine, Channel and Select case.

**Problem Solving:**

1. Program based on string operation (e.g., permutation of given string)
2. Program based on array (e.g., to find the smallest missing positive integer number from given slice/array)

**Overall Feedback on earlier rejected profiles:**

1. Golang - Project experience is missing in the profile. Need related Golang exp.
2. Python exp is not relevant.
3. Golang concepts, Rest API - Missing
4. Slice, Channel, Goroutine - Theoretical knowledge will not suffice, need to explain/answer scenario-based questions.
5. Problem Solving - Simple program, make API call to 2 different services, get the data, aggregate it. Call 3rd service to post the data.

# Customer Expectation

1. Theoretical knowledge will not suffice, need to explain/answer practical scenario.
2. He / she should be able to explain project in a very good way and check is it really Golang project asking some question based on project.
3. Problem Solving Example,
   1. make API call of 2 diff services, get the data, and aggregate it. And call post API 3 services with aggregate data
   2. Here syntax is ok, but logic need to evaluate and need to check is it following coding standards (separate function, commenting, modularity etc.)
   3. Look for optimization to use go routine to call 2 diff API of services.
   4. Rest standards etc.
4. Problem Solving Example
   1. Given string find the space and next to space do the char to uppercase irrespective of existing case of char.
   2. Expectation solution first
   3. If he did the solution, then ask without using in build function.
5. Problem Solving Example
   1. Find the second min number in each array/slice.
6. Problem Solving Example
   1. Stack or Queue implementation in a Golang
   2. Here he/she should use methods with pointer receiver.

# Problem 1:

accepts input as text and provides Output as Top ten most used words and times of occurrence.

Occurrences should be in descending order

Sample Input string: - input:= "I am Manish Manish is good person Manish like cricket Manish like music he like mango he cares others"

Sample Output: - [{Manish 4} {like 3} {he 2} {cares 1} {others 1} {music 1} {I 1} {mango 1} {am 1} {person 1}]

Solution link: <https://go.dev/play/p/FrmDj-wWy-9>

Evalution Criteria and Score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No. | Expectation |  | Golang Concepts Used | Score |
| 1 | Define separate function and passed input text |  | Best Practice | 1 |
| 2 | Split the input based on spaces |  | String's function | 1 |
| 3 | find out the words and its count |  | Map, for loop, range | 1 |
| 4 | sort the output based on count |  | Own Sort interface implementation | 2 |
| 5 | Print the top 10 most used words with its count  **Note: if the words are less than 10 then print all available words.** |  | Slice | 1 |
| 6. | If we find the word count same, then sort word as well alphabetical (Optional) |  |  | 2 |

Problem 2:

1. /\*write a program which will print odd, even numbers from 1-10. here odd numbers will be printed by go routine 1(job1) and even numbers will be printed by go routine 2(job2).

we are expecting to use 2 go routines (odd and even) and for blocking/unblocking use suitable concepts from Golang. At most only 2 Goroutine need to spawn. So if we did the calculations how much Goroutine was spawn that result should be 2(in total 2).

Condition

1. Go routine Job will be just print the number whatever number they will receive.
2. To check no is even or odd you need to add this login in main function itself and pass same to respective go routine
3. At then end only 2 Go Routine should be there

Input:-

numbers:= []int{1,2,3,4,5,6,7,8,9,10}

output should be in sequence like below.

Job1... 1  
Job2... 2  
Job1... 3  
Job2... 4  
Job1... 5  
Job2... 6  
Job1... 7  
Job2... 8  
Job1... 9  
Job2... 10 \*/

Solutions:- we are expecting to use go routines with channel

Or if there is other solution welcome other than time.sleep() for blocking

Solution using Channel <https://play.golang.com/p/LaZ5y8uvVt5>

<https://go.dev/play/p/nXtY8OCu7ky>

Evolutions

|  |  |  |  |
| --- | --- | --- | --- |
| Sr no | Expectations | Golang Concept | Score |
| 1 | Define separate function for to print odd and to print even number | Best Practice | 1 |
| 2 | Define WaitGroup or Chan | Waitgroup or chan | 1 |
| 3 | Use Login to wait the loop till go routines finished | Chan blocking or Wait func from Waitgroup | 1 |
| 4 | Expected output in a sequence |  | 2 |

**Handel multiple channels**

Write/complete the code to handle response from multiple channels. Using for select.

package main

import (

"fmt"

"math/rand"

"time"

)

func GetRandomInt(rng int)int{

return rand.Intn(rng)

}

func main() {

ch1 := make(chan int)

ch2 := make(chan string)

go func(ch chan int){

for i:=0;i<5;i++{

ch <- GetRandomInt(1000)

time.Sleep(time.Millisecond \* time.Duration(GetRandomInt(5)))

}

close(ch)

}(ch1)

go func(ch chan string){

for i:=0;i<15;i++{

ch <- fmt.Sprintf("ittrID-%d",i)

time.Sleep(time.Millisecond \* time.Duration(GetRandomInt(5))) }

close(ch)

}(ch2)

// write your code to handle channel o/p

}

Solution: <https://play.golang.com/p/64CtX98Y0E5>

|  |  |  |  |
| --- | --- | --- | --- |
| Sr no | Expectations | Golang Concept | Score |
| 1 | Define separate function for to print odd and to print even number | Best Practice | 1 |
| 2 | Use of for select. | Use of for select | 2 |
| 3 | Expected output |  | 2 |

Singly Linked list Implementation

package main

import "fmt"

func main() {

    node := NewNode(10, NewNode(30, NewNode(50, nil)))

    PrintNode(node)

}

type Node struct {

    data int

    next \*Node

}

func NewNode(data int, next \*Node) \*Node {

    var v Node

    v.data = data

    v.next = next

    return &v

}

func PrintNode(node \*Node) {

    for node != nil {

        fmt.Println(node.data)

        node = node.next

    }

}

Some Programs Implementation and it’s link

1. Write prog to print numbers from slice using  
   a. Single goroutine & channel  
   b. 2 goroutines & 2 channels

[Go Playground - The Go Programming Language](https://go.dev/play/p/HlMG7uuiSYW)

// Program to print numbers from slice using 2 goroutines and channels

package main

import (

"fmt"

)

func main() {

var intSlice = []int{91, 42, 23, 14, 15, 76, 87, 28, 19, 95}

chOdd := make(chan int)

chEven := make(chan int)

go odd(chOdd)

go even(chEven)

for \_, value := range intSlice {

if value%2 != 0 {

chOdd <- value

} else {

chEven <- value

}

}

}

func odd(ch <-chan int) {

for v := range ch {

fmt.Println(v)

}

}

func even(ch <-chan int) {

for v := range ch {

fmt.Println(v)

}

}

Write a function to return min and Max from slice

// program to return min, max from array

[Go Playground - The Go Programming Language](https://go.dev/play/p/U6ev2DhOIR1)

package main

import "fmt"

func main() {

var a = [5]int{11, -4, 7, 8, -10}

min, max := findMinAndMax(a)

fmt.Println("Min: ", min)

fmt.Println("Max: ", max)

}

func findMinAndMax(a [5]int) (min int, max int) {

min = a[0]

max = a[0]

for \_, value := range a {

if value < min {

min = value

}

if value > max {

max = value

}

}

return min, max

}

 What is linkedList ? What is difference between array and linkedList ?  
Write a program to add node in linkedList

[Go Playground - The Go Programming Language](https://go.dev/play/p/h54Ix-xPG_7)

// Program to add node in linked list

package main

import "fmt"

type Node struct {

value int

next \*Node

}

func NewNode(value int, next \*Node) \*Node {

var n Node

n.value = value

n.next = next

return &n

}

func TraverseLinkedList(head \*Node) {

temp := head

for temp != nil {

fmt.Printf("%d ", temp.value)

temp = temp.next

}

fmt.Println()

}

func AddNodeAtEnd(head \*Node, data int) \*Node {

if head == nil {

head = NewNode(data, nil)

return head

}

temp := head

for temp.next != nil {

temp = temp.next

}

temp.next = NewNode(5, nil)

return head

}

func main() {

head := NewNode(30, NewNode(10, NewNode(40, NewNode(40, nil))))

fmt.Printf("Input Linked list is: ")

TraverseLinkedList(head)

AddNodeAtEnd(head, 5)

fmt.Printf("After adding node at end, linked list is: ")

TraverseLinkedList(head)

}

1. Write a all permutation of given string s:=”abc” .

Output: abc, acb,cab,cba,bac,bca

package main

import "fmt"

func main() {  
  a := "abc"  
  result := gen(a)  
  fmt.Println("Combinations are : ", result)

}

func gen(a string) []string {  
  result := []string{}

  genhelp([]byte(a), 0, &result)  
  return result  
}

func genhelp(s []byte, start int, result \*[]string) {  
  if start == len(s) {  
    \*result = append(\*result, string(s))  
    return  
  }

  for i := start; i < len(s); i++ {  
    s[start], s[i] = s[i], s[start]  
    genhelp(s, start+1, result)  
    s[start], s[i] = s[i], s[start]  
  }  
}

1. Write a program to find out the smallest missing positive integer from unsorted list/slice. Slice may contain negative numbers as well.

Here you need to use one goroutine and one channel.

Eg. S:=[]int{-1,1,8,9,3,4,5,6} o/p:- 2

<https://go.dev/play/p/YmOMEz2wnFj>

package main

import (  
  "fmt"  
  "sort"  
)

// Write a program to find out the smallest missing positive integer from unsorted list/slice. Slice may contain negative numbers as well.

// Eg. S:=[]int{-1,1,3,4,5,6} o/p:- 2

func main() {

  list := []int{-1, 1, 2, 3, 4, 7, 6, 5, -4, -3, 8, 10}

  sort.Ints(list)

  fmt.Println(list)  
  check := 0  
  for i := 0; i < len(list); i++ {  
    check = list[i] + 1  
    if check <= 0 {  
      continue  
    } else {  
      if i+1 < len(list) {  
        if list[i+1] == check {  
          continue  
        } else {  
          fmt.Println(check)  
          break  
        }  
      } else {  
        fmt.Println(check)  
      }  
    }

  }  
}

1. We have given 2 string n,n+1. Your task is found out the extra char which is in second string. Solve same problem using 1 Go routine and 1 chan.

Eg. S:=”abc”, s1:=”abcd” , so here d is extra char.

1. Write a program to find minimum number of coins that makes a given value

[Find minimum number of coins that make a given value - GeeksforGeeks](https://www.geeksforgeeks.org/find-minimum-number-of-coins-that-make-a-change/)

Eg. ***Input:****coins[] = {25, 10, 5}, V = 30****Output:****Minimum 2 coins required We can use one coin of 25 cents and one of 5 cents*

***Input:****coins[] = {9, 6, 5, 1}, V = 11****Output:****Minimum 2 coins required We can use one coin of 6 cents and 1 coin of 5 cents*

1. *Reverse an array in a group of given size*

[Reverse an Array in groups of given size - GeeksforGeeks](https://www.geeksforgeeks.org/reverse-an-array-in-groups-of-given-size/?ref=lbp)

*Eg.* ***Input:****arr[] = [1, 2, 3, 4, 5, 6, 7, 8, 9], K = 3****Output:****3, 2, 1, 6, 5, 4, 9, 8, 7*

***Input:****arr[] = [1, 2, 3, 4, 5, 6, 7, 8], K = 5****Output:****5, 4, 3, 2, 1, 8, 7, 6*

***Input:****arr[] = [1, 2, 3, 4, 5, 6], K = 1****Output:****1, 2, 3, 4, 5, 6*

***Input:****arr[] = [1, 2, 3, 4, 5, 6, 7, 8], K = 10****Output:****8, 7, 6, 5, 4, 3, 2, 1*

1. *Write a program which will print only unique value from slice. Go this you need to use one go routine and one chan.*

*Arr:=[1,1,1,2,3,4,5,5,3,4,6,7,8,9,10,11,11,12,13]*

1. Write a below application. Student Create and Get operations
2. POST /students

{“name”:”abc”,”id”:1}

Success 201 conflict 409

1. Get /students/{id}

200 ok resp {“name”:”abc”,”id”:1}

404 Not found

1. Write a program which will do the upper case if first character whenever space occurred in a given sentence.

Eg:= “i am divya. working in encorra invoation lab. I am good in golang”

o/p:= “i Am Divya. Working In Encorra Invoation Lab. I Am Good In Golang”

Cloud-native application reference architecture includes a set of technologies to build and run scalable applications in public, private, and hybrid clouds. With techniques that include containers, service meshes, and microservices

commercetools, is a cloud-based headless commerce platform that provides APIs to power e-commerce sales and similar functions for large businesses.

New Role:- Senior Lead Engineer

Already managing the team. Following the process. Taking the interviews(for senior and arch role as well).

Got the good feedback from client and I am very focused on the process , work and quality as well.

I worked in Brillio in Identity, MHP, Terranova Partner platform migration and now Content Reputation. Here because of contract / budget issue not got much chance to stick with one team understand more in details. Still tried to grap the new things early and worked on same.

Current pr

Plan for future

* 1. Need to learn one more backend lang which Node JS
  2. Also plan to learn one front end lang which is React JS
  3. Both lang which planning to learn from our internal tool SkillSoft Percipio
  4. Also planning to the one of cloud certification mostly AWS and because of that started attending free webinar session from AWS.