1. **Go program to do wordwise reverse of a string.**

```go
package main

import (
    "fmt"
    "strings"
)

func main() {
    var result string
    var res string
    var str1 []string
    str := "welcome to go"
    str1 = strings.Split(str, " ")
    for _, v := range str1 {
        result = string(v) + result
    }
    for _, u := range result {
        res = string(u) + res
    }
    fmt.Println(res)
}
```

Output:

```
PROBLEMS 41   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell + v  ^ X

PS D:\go\sample\demo\strings> go run reversestr.go
emoclewotog
PS D:\go\sample\demo\strings>
```

2. Go program to print average of numbers using array.

```go
package main

import "fmt"

func main() {
    var num [100]int
    var temp, sum, avg int
    fmt.Print("Enter number of elements: ")
    fmt.Scanln(&temp)
    for i := 0; i < temp; i++ {
        fmt.Print("Enter the number : ")
        fmt.Scanln(&num[i])
        sum += num[i]
    }

    avg = sum / temp
    fmt.Printf("The Average of entered %d number(s) is %d", temp, avg)
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell

PS D:\go\sample\demo\strings> go run avg.go
Enter number of elements: 5
Enter the number : 2
Enter the number : 22
Enter the number : 15
Enter the number : 16
Enter the number : 23
The Average of entered 5 number(s) is 15
PS D:\go\sample\demo\strings>
```

3. Hash table example using go lang.

```go
package main

import (
    "fmt"
)

func main() {
    var country map[int]string
    country = make(map[int]string)
    country[1] = "Vikas"
    country[2] = "vinita"
    country[3] = "vinutha"
    country[4] = "gowtham"
    country[5] = "shashi"
    country[6] = "triambak"
    for i, j := range country {
        fmt.Printf("Key: %d, Value: %s\n", i, j)
    }
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell + v  ^ X

PS D:\go\sample\demo\strings> go run hash.go
Key: 1, Value: Vikas
Key: 2, Value: vinita
Key: 3, Value: vinutha
Key: 4, Value: gowtham
Key: 5, Value: shashi
Key: 6, Value: triambak
PS D:\go\sample\demo\strings>
```

4. Go program to print full pyramid.

```go
package main

import "fmt"

func main() {
    var rows int
    var k int = 0
    fmt.Print("Enter number of rows :")
    fmt.Scan(&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:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell

PS D:\go\sample\demo\strings> go run pyramid.go
Enter number of rows :5
      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * * *
PS D:\go\sample\demo\strings>
```

5. **Go program to swap 2 numbers without using temporary variable.**

```go
package main

import "fmt"

func main() {
    fmt.Print("Enter first number : ")
    var first int
    fmt.Scanln(&first)
    fmt.Print("Enter second number : ")
    var second int
    fmt.Scanln(&second)
    first = first - second
    second = first + second
    first = second - first
    fmt.Println("First number :", first)
    fmt.Println("Second number :", second)
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell + v  ^ X

PS D:\go\sample\demo\strings> go run swap.go
Enter first number : 12
Enter second number : 13
First number : 13
Second number : 12
PS D:\go\sample\demo\strings>
```

6. Go program to find area of rectangle and square.

```go
package main

import "fmt"

var area int // Variable declare outside the main function

func main() {
    var l, b int //Declaration of multiple Variables
    fmt.Print("Enter Length of Rectangle : ")
    fmt.Scan(&l)
    fmt.Print("Enter Breadth of Rectangle : ")
    fmt.Scan(&b)
    area = l * b
    fmt.Println("Area of Rectangle : ", area) //move to new line

    fmt.Print("Enter Length of Square : ")
    fmt.Scan(&l)
    area = l * l
    fmt.Print("Area of Square : ", area)
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell + v  ^ X

PS D:\go\sample\demo\strings> go run area.go
Enter Length of Rectangle : 15
Enter Breadth of Rectangle : 15
Area of Rectangle :  225
Enter Length of Square : 20
Area of Square : 400
PS D:\go\sample\demo\strings>
```

7. Go program to find mx records record of domain.

```go
package main

import (
    "fmt"
    "net"
)

func main() {
    mxrecords, _ := net.LookupMX("instagram.com")
    for _, mx := range mxrecords {
        fmt.Println(mx.Host, mx.Pref)
    }
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell + v  ^ X

PS D:\go\sample\demo\strings> go run mx.go
mxa-00082601.gslb.pphosted.com. 10
mxb-00082601.gslb.pphosted.com. 10
PS D:\go\sample\demo\strings>
```

8. Bubble sort using channel.

```go
package main

import "fmt"

func bubbleSort(array []int, c chan []int) []int {
    for i := 0; i < len(array)-1; i++ {
        for j := 0; j < len(array)-i-1; j++ {
            if array[j] > array[j+1] {
                array[j], array[j+1] = array[j+1], array[j]
            }
        }
    }
    c <- array
    return array
}
func printOutput(arr []int) []int {
    fmt.Println(arr)
    return arr
}
func main() {
    c := make(chan []int, 7)
    array := []int{11, 14, 3, 8, 18, 17, 43}
    array2 := []int{3, 12, 16, 7, 19, 8}
    array3 := []int{9, 8, 7, 6, 5}
    go bubbleSort(array, c)
    go bubbleSort(array2, c)
    go bubbleSort(array3, c)
    sorted := <-c
    sorted1 := <-c
    sorted2 := <-c
    fmt.Println(sorted)
    fmt.Println(sorted1)
    fmt.Println(sorted2)
}
```

Output:

```
PROBLEMS 40   OUTPUT   TERMINAL   DEBUG CONSOLE                    powershell

PS D:\go\sample\demo\strings> go run channel.go
[3 8 11 14 17 18 43]
[5 6 7 8 9]
[3 7 8 12 16 19]
PS D:\go\sample\demo\strings>
```

9. Go program to find standard deviation.

```go
package main

import (
    "fmt"
    "math"
)

func main() {
    var num [10]float64
    var sum, mean, sd float64
    fmt.Println("******  Enter 10 elements  *******")
    for i := 1; i <= 10; i++ {
        fmt.Printf("Enter %d element : ", i)
        fmt.Scan(&num[i-1])
        sum += num[i-1]
    }
    mean = sum / 10

    for j := 0; j < 10; j++ {
        sd += math.Pow(num[j]-mean, 2)
    }
    sd = math.Sqrt(sd / 10)

    fmt.Println("The Standard Deviation is : ", sd)
}
```

Output:

```
PS D:\go\sample\demo\strings> go run sd.go
******  Enter 10 elements  *******
Enter 1 element : 2
Enter 2 element : 4
Enter 3 element : 6
Enter 4 element : 8
Enter 5 element : 10
Enter 6 element : 12
Enter 7 element : 14
Enter 8 element : 16
Enter 9 element : 18
Enter 10 element : 20
The Standard Deviation is :  5.744562646538029
PS D:\go\sample\demo\strings>
```

10. Go program for writing a json file.

```go
package main

import (
    "encoding/json"
    "io/ioutil"
)

type Salary struct {
    Basic, HRA, TA float64
}

type Employee struct {
    FirstName, LastName, Email string
    Age                        int
    MonthlySalary              []Salary
}
func main() {
    data := Employee{
        FirstName: "Vikas",
        LastName:  "S",
        Email:     "vikas@gmail.com",
        Age:       25,
        MonthlySalary: []Salary{
            Salary{
                Basic: 115000.00,
                HRA:   15000.00,
                TA:    12000.00,
            },
            Salary{
                Basic: 16000.00,
                HRA:   25000.00,
                TA:    21000.00,
            },
            Salary{
                Basic: 170000.00,
                HRA:   50000.00,
                TA:    22000.00,
            },
        },
    }
    file, _ := json.MarshalIndent(data, "", " ")

    _ = ioutil.WriteFile("test.json", file, 0644)
}
```

Output:

strings > {} test.json > ...

```
{
 "FirstName": "Vikas",
 "LastName": "S",
 "Email": "vikas@gmail.com",
 "Age": 25,
 "MonthlySalary": [
  {
   "Basic": 115000,
   "HRA": 15000,
   "TA": 12000
  },
  {
   "Basic": 16000,
   "HRA": 25000,
   "TA": 21000
  },
  {
   "Basic": 170000,
   "HRA": 50000,
   "TA": 22000
  }
 ]
}
```

PROBLEMS 43 OUTPUT TERMINAL DEBUG CONSOLE powershell

```
PS D:\go\sample\demo\strings> go run json.go
PS D:\go\sample\demo\strings>
```