

Proyecto 1

Es bien sabido que la necesidad de las cosas mueve el mundo, una teoría muy arriesgada fue la que tomo el explorador noruego **Thor Heyerdahl**; Era el año 1947 y este joven aventurero noruego, se dirigía con rumbo a Polinesia para probar una arriesgada teoría: que los antiguos habitantes de Sudamérica habían sido capaces de cruzar el océano Pacífico para colonizar los archipiélagos localizados a miles de kilómetros de distancia de su tierra natal.

Para probar su hipótesis, Heyerdahl fue a Perú y comenzó a construir una balsa basada en descripciones e ilustraciones españolas de embarcaciones locales durante la época de la conquista. Era una embarcación hecha de fibras vegetales y madera, la cual poseía un mástil principal con una vela cuadrada, junto con una choza como cabina para la tripulación. Era el día 28 de abril de 1947, y con una tripulación de 6 hombres y un loro, emprendieron un viaje desde el puerto del Callao para probar la factibilidad de tal viaje transpacífico. Como nombre de bautizo para su embarcación, eligió el de "**Kon-Tiki**", que habría sido una antigua denominación del dios Viracocha, cuya interpretación de su imagen fue dibujada como símbolo en la vela de su balsa.

Luego de 101 días en el mar, consumiendo tubérculos, cocos, raciones de supervivencia y peces que capturaron en el viaje, los 6 miembros de la expedición lograron llegar al arrecife de Raroia en la Polinesia francesa, ¡a una distancia de 6.980 kilómetros! Su viaje fue un éxito y Thor Heyerdahl se convirtió en una celebridad mundial. Su libro sobre la expedición fue publicado en 1948 y fue un best-seller de ventas, pero la comunidad científica se mantuvo escéptica de sus teorías sobre el poblamiento sudamericano de Polinesia.¹

Este explorador fue capaz de sumergirse en mares desconocidos, y en esta oportunidad sera nuestro caso, supongamos que estamos en una habitación solo con accesos a un teclado y un monitor, solo podemos ver algunas cosas de internet no todo, y es necesario afrontar un reto no tan grande como el que vimos anteriormente pero si necesario para sobrevivir. Nuestro carcelero, a si olvide decirlo están atrapados, detrás de 4 puertas, cada una de ellas representa un reto, y debemos superarlos para alcanzar la libertad, pero bueno ya me perdí en que estaba? A si ya lo recordé, solo cuentas con poco tiempo y los recursos antes mencionados, su carcelero es lo suficientemente inteligente para saber si se copian, o si usan recursos no permitidos tales como sort, reflect y funciones de orden superior, el tiene reglas estas son:

- Puedes usar cualquier tipo de estructura de dato, pero el recomienda que lean acerca de los slices y que "response := make([]<tipo_dato>,0)" es igual a response := []<tipo_dato>
- en ocasiones los string se vuelven int, es loco pero cierto string(<valor_int>) es una buena forma de resolver ciertas cosas.
- Cada ejercicio es una función que sera llamada desde el main.
- go mod init <name_modulo> es una buena practica, [Aquí entre nos se hace en la raíz del proyecto por consola].
- Visual studio code es buen editor, pero si pueden usen intelliJ les solucionara mucho la vida y es bueno para el dia a dia.

¹ [De Peru a la Polinesia](#)

- El carcelero es buena gente siempre y cuando tengan algo, si no tienen nada no pidan ayuda, [no son los unicos prisioneros]
- No es necesario que se mate haciéndolo bonito, con que se vea por consola es suficiente.
- Si usa otro lenguaje debe estar bien preparado para responder muchas preguntas, y si una no responde pues lo demas quedara anulado, le recomiendo que no lo haga, use go. [pueden hacerlo en otro lenguaje, bajo pena de uso de lo antes mencionado]
- Las funciones globales empiezan con Mayuscula. Puede hacer todo en un solo archivo donde esta el main, pero procure hacerlo por package, no es difícil solo cree una carpeta “name”, cuando importe la función en el main es name.FunctionName
- La primera semana no hay pregunta de ningún tipo.

Una vez aclarada las reglas veamos que se desea, no se asuste respire Ud puede hacer esto, el carcelero lo hizo en 4 horas Ud contara con 1 mes aproximadamente. Ud debe resolver 2 ejercicios relacionados a los strings y 2 a los arrays.

Strings:(7 puntos, 2 y 5 respectivamente)

- Crear una función palindrome, esta recibe un string y valida si es palindrome o no, debe retornar un booleano si es palindrome o no, ej:
func IsPalindrome(w string) bool, IsPalindrome(“ana”) => true; IsPalindrome(“raro”) =>false
- Crear una función que valide si una palabra esta dentro de una frase, recibe dos string la palabra y la frase y retorna un booleano, eje:
func ExistWord(w, fragment string) bool; ExistWord(“ana”, “adios bonitana despues de ti”) =>true; ExistWord(“arpa”, “adios bonitana despues de ti”) => false.

Arrays: (13 puntos , 6 y 7 respectivamente)

- Crear una función que dado un array extraiga las letras repetidas contiguas, y retorne un array de arrays con los resultados, es decir:

```
func Extract(arr []string) [][]string;
arr := []string{"a", "a", "c", "c", "s", "x", "x", "x", "x", "a"}
Extract(arr) =>[[a a] [c c] [x x x x]]
```

Nota: Solo se deben tomar los contiguos, fíjese que no se toma ni la s ni la ultima a ya que no posee elementos contiguos, en el siguiente ejemplo se ve mejor

```
arr := []string{"a", "z", "12"}
Extract(arr) =>[]
```

El resultado es vacío porque no existen elementos repetidos contiguos entre si.

- Crear una función que sea capaz de generar una semilla entera, es decir la función debe recibir dos arrays de uno con las letras para generar la semilla y otro con las letras que pueden aportar semilla es decir;

tenemos el array["a","c","c"] que representa a las letras que generan semilla y el array["y","a","x"], para este caso la única letra que puede generar semilla es la letra a, que se encuentra en ambos array, el peso de la letra "a" es de 1, lo que representa su índice en el segundo array, la formula para calcular la semilla es " Σ numero de apariciones" + " \prod numero de apariciones", para este caso existe una sola aparición lo que se traduce en "1"+"1". =2 sería la semilla generada.

Veamos otro ejemplo mas completo, tenemos el array de letras ["a","c","x","a"] y cambiamos el array de los pesos a ["y","a","x"], la operación arroja como resultados $(1+3+1) + (1*3*1) = 8$

La función debe recibir:

```
func CreateSeed(s, p []string) int
```

Debe tomar en consideración que la semilla no puede ser un numero menor o igual a 0.

Si existe algún tipo de duda con lo aquí planteado tienen hasta el martes para aclarar solo dudas de enunciado.

Bonus extra:

- usar Git.
- Usar Apuntadores.
- Crear interfaz grafica si y solo si los ejercicios funcionan correctamente.
- Almacenar los resultados en archivos por separado.
- Leer los casos de prueba desde archivos.

Enjoy :)

