Moisés Fernández Zárate A01197049

Programación avanzada (Gpo 2)

Linux Service Challenge

Para esta práctica solucioné el problema Player de la página CodeChef con el código RRPLAYER. El link de la página es: <https://www.codechef.com/problems/RRPLAYER>

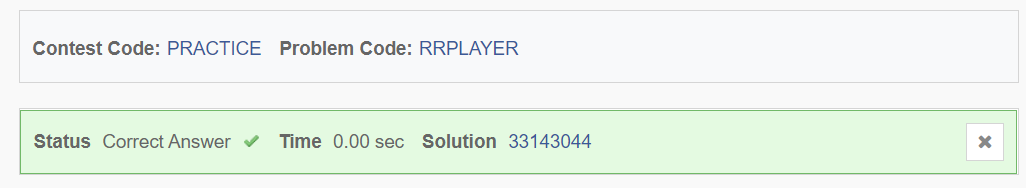
### **Explicación del problema**

El problema consiste en que los 3 críticos musicales más influyentes de San Petersburgo quieren escuchar el nuevo álbum de su banda favorita. Después de una discusión deciden escuchar las canciones en un orden aleatorio. Hay N canciones en el álbum, cada canción es aleatoria, lo cual quiere decir que cada canción tiene la misma probabilidad de ser escogida. Después de cada canción, la siguiente canción que van a escuchar es escogida al azar, independientemente de las canciones que ya escucharon. Lo que pide el problema es regresar cuál es el número esperado de canciones que los 3 críticos tienen que escuchar hasta haber escuchado cada canción del álbum por lo menos una vez.

### **Explicación de la solución**

Lo que hice para resolver este problema fue que en cada caso, hacía un for loop de 1 a n, n siendo la cantidad de canciones en el álbum, mi variable para guardar la respuesta empezaba en cero y le voy sumando la división de n entre i para el cálculo de la respuesta porque después de haber escuchado una canción, va subiendo la probabilidad de terminar el álbum con la cantidad de canciones en total divididas por las canciones que han sido reproducidas hasta terminar todas las canciones del álbum.

### **Tiempo de la solución**



### **Tamaño del archivo**

Archivo del servicio



Archivo de la solución



### **Proceso para transformarlo en un servicio**

Para realizar esta práctica utilicé del material de apoyo de la maestra que está en Blackboard, un link que contiene información para escribir un daemon en Linux utilizando GCC. Después de seguir los pasos de la página y generar el código como se explica ahí, transformé mi solución al problema de CodeChef a que utilizara archivos para la implementación del servicio y luego llamé a la función de ese archivo para que se ejecute en el servicio cada determinado tiempo, que en mi caso establecí que fuera cada 10 segundos.