

Parcial Funcional: Haskellpark

Disclaimer: En este parcial se evaluarán los conceptos de aplicación parcial, composición, reutilización, recursividad y orden superior para el modelado del mismo. En caso de no estar presentes estos conceptos se descontará puntaje.



Modelamos una aplicación de parques de diversiones donde registramos sus atracciones, de las cuales conocemos su nombre, altura mínima requerida para la persona que ingrese medida en centímetros, duración en minutos, una serie de opiniones que le da la gente ("entretenida", "veloz", "un embole", etc) y si está en mantenimiento o no lo cual permite el acceso por parte de las personas. Un operador determina si el entretenimiento requiere atención y lo pasa a su estado de mantenimiento. En algún momento pasan los técnicos y asignan las reparaciones, que tiene una duración determinada en días y el trabajo que se realiza.

Punto 1: Más bueno que ... (2 puntos)

Queremos saber qué tan buena es una atracción. Para eso utilizamos un sistema de scoring que tiene un modo muy particular para calcularlo: Si la atracción tiene una duración prolongada (es decir que dura más de 10 minutos) valen 100 puntos. Si no es así, pero tiene menos de 3 órdenes de reparaciones el puntaje es 10 puntos por cada letra del nombre más 2 puntos por cada opinión que tiene. Caso contrario es 10 veces la altura mínima requerida.

Punto 2: Iguana fissss (4 puntos)

Los técnicos tienen diversos tipos de trabajo que pueden desarrollar en cada reparación sobre las atracciones. Algo muy importante a tener en cuenta es que luego de que realizan cualquier trabajo siempre ocurren dos cosas: se elimina la última reparación de la lista (no importa cual fuere) y se verifica que no tenga reparaciones pendientes. Si quedan pendientes debe mantener el indicador que está en mantenimiento, caso contrario no. Los posibles trabajos son :

ajusteDeTornillería que como le refuerza aún más estructura a la atracción, prolonga su duración en 1 minuto por cada tornillo apretado pero no pudiendo superar los 10 minutos porque no es rentable. Es decir que si una atracción dura 3 minutos y ajusta 4 tornillos la misma pasa a durar 7 minutos. Pero sí una atracción dura 8 minutos y el técnico logra apretar 5 tornillos pasa a durar solamente 10 minutos.

engrase que vuelve más veloz al entretenimiento, por lo tanto aumenta en 0,1 centímetros la altura mínima requerida por cada gramo de grasa utilizada en el proceso y le agrega la opinión "para valientes". La cantidad de grasa requerida puede variar según el criterio del técnico.

mantenimientoElectrico repara todas las bombitas de luz y su cableado. Como es un lavado de cara y una novedad para la gente, solo se queda con las dos primeras opiniones y el resto las descarta.

mantenimientoBásico que consiste en ajustar la tornillería de 8 tornillos y hacer un engrase con 10 gramos de grasa.

Parcial Funcional: Haskellpark

Punto 3: ¿Qué oooooonda este parque? (3 puntos)

Esa me da miedito

Queremos saber si una atracción *meDaMiedito*, esto implica que alguna de las inspecciones que se le hicieron le asignó más de 4 días de mantenimiento.

Acá cerramos...

Cerramos una atracción si la sumatoria de tiempo de las reparaciones pendientes para dicha atracción es de 7 días.

Disney no existis

Tenemos que determinar *disneyNoExistis* para un parque. Esto ocurre cuando todas las atracciones de nombre cheto (con más de 5 letras) no tienen reparaciones pendientes.

En este punto no puede utilizar funciones auxiliares ni recursividad, solo composición y aplicación parcial.

Punto 4: Reparaciones peolas (2 puntos)

Una atracción tiene reparaciones peolas si luego de cada una está más buena, esto implica que luego de hacer el trabajo de cada reparación el puntaje mejora con respecto a la reparación previa.

En este punto debe resolver exclusivamente con recursividad

Punto 5: Manny a la obra (2 puntos)

Queremos modelar un proceso que realice los trabajos de las reparaciones pendientes sobre una atracción. Se pide que además muestre un ejemplo de cómo podría evaluar por consola el proceso para cada una de las actividades resueltas en el punto anterior.

En este punto NO puede utilizar recursividad

Punto 6: Estoy cansado jefe... (1 puntos)

Si una atracción tiene una cantidad infinita de trabajos, ¿sería posible obtener un valor computable para la función del punto anterior? ¿Qué ocurriría con una lista de trabajos infinita en el punto 4? Justifique sus respuestas relacionándolo con un concepto visto en la materia.

13,50-14 puntos = 10 | > 12 - 13,50 puntos = 9 | > 11 - 12 puntos = 8 | > 10 - 11 puntos = 7 | > 9 - 10 puntos = 6 | 8 - 9 puntos = revisión | resto: desapueba