

## Jugando cartas con un funcional - Parte B

**(a)** Escriba una función *score\_challenge* y *officiate\_challenge* que son parecidas a las de la parte A, excepto que el As puede tener un valor de 1 u 11 y *score\_challenge* debe retornar el menor (en otras palabras el mejor) *score* posible. Nota: recuerde que el juego termina si la suma excede el goal, en este caso hay que calcular las posibilidades.

**(b)** Escriba *careful\_player*, que toma una lista de cartas y un *goal*, y retorna una *move-list* tal que si llamamos a *officiate* con esa *card-list*, el *goal*, y el *move-list* tiene el siguiente comportamiento:

- El valor de las *held-cards* nunca excede el *goal*.
- Se va a llevar a cabo *drawing* de una carta siempre que el goal sea mayor en 10 que el valor de las *held-cards*.
- Si el *score* alcanzado es 0, no deben haber más movimientos.
- Si es posible alcanzar un *score* de 0 haciendo *discarding* seguido de un *drawing*, entonces debe ser hecho de esa forma. Nótese que *careful\_player* debería de poder ver la siguiente carta (que en algunos juegos sería como trampa).
- Este problema **(b)** no es una continuación del anterior. Por lo tanto en este problema se considera al As solamente con valor de 11.

### Notas finales

- Este proyecto debe ser resuelto en parejas. Pero hay una condición: ambas personas deben dominar el código, de lo contrario habrá una rebaja de puntos.
- Crear un solo archivo con la solución de la tarea.
- Hacer los casos de prueba.
- Fecha de entrega: \_\_\_\_\_.