

## Memory

Un juego llamado Memory se juega con 50 cartas. Cada carta tiene una de las letras de la A a la Y (ASCII 65 a 89) impresa en su cara no visible, de modo que cada letra aparece exactamente en dos tarjetas. Las cartas se barajan en un orden aleatorio y se reparten boca abajo sobre la mesa.

Jack juega al juego poniendo dos cartas boca arriba para que las letras sean visibles. Por cada una de las 25 letras, Jack recibe un caramelo la primera vez que ve ambas copias de la letra en las dos cartas boca arriba a la vez. Por ejemplo, la primera vez que Jack da la vuelta a las dos cartas que contienen la letra M, recibe un caramelo. Independientemente de si las letras son iguales o no, Jack luego coloca ambas cartas boca abajo de nuevo. El proceso se repite hasta que Jack recibe 25 caramelos, uno por cada pareja con la misma letra.

Debes implementar una función **play** que juegue la partida. Tu implementación debe llamar a la función **faceup (C)**, que ya está implementada como parte del grader. C es un número entre 1 y 50 que indica una carta en particular que quieres colocar boca arriba. La carta no debe estar boca arriba en el momento de la llamada. **faceup (C)** devuelve la letra que está impresa en la carta C.

Las llamadas a **faceup (C)** van por pares. La primera llamada levanta una carta y la deja boca arriba. La segunda llamada levanta otra carta y, tras procesar su valor en relación a la primera carta, vuelve a colocar ambas boca abajo.

Tu función **play** solo puede terminar una vez que Jack haya recibido los 25 caramelos. Está permitido hacer llamadas a **faceup (C)** incluso después del momento en el que Jack reciba el último caramelo. Tu puntuación se calculará en función del número de llamadas a **faceup (C)** por parte de **play**.

### Ejemplo de secuencia parcial del juego

Llamada	Valor devuelto	Explicación
faceup(1)	'B'	La carta 1 contiene la letra 'B'. La carta 1 queda levantada.
faceup(7)	'X'	La carta 7 contiene la letra 'X'. Tras esta

		llamada, el grader pone boca abajo las cartas 1 y 7.
faceup(7)	'X'	La carta 7 contiene la letra 'X'. La carta 7 queda levantada.
faceup(15)	'O'	La carta 15 contiene la letra 'O'. Tras esta llamada, el grader pone boca abajo las cartas 7 y 15.
faceup(7)	'X'	La carta 7 contiene la letra 'X'. La carta 7 queda levantada.
faceup(50)	'X'	La carta 50 contiene la letra 'X'. Tras esta llamada, el grader pone boca abajo las cartas 7 y 50. Las letras coinciden, así que Jack recibe un caramelo.
faceup(7)	'X'	La carta 7 contiene la letra 'X'. La carta 7 queda levantada.
faceup(50)	'X'	La carta 50 contiene la letra 'X'. Tras esta llamada, el grader pone boca abajo las cartas 7 y 50. Aunque las letras coinciden, Jack no recibe ningún caramelo, ya que ya recibió el caramelo correspondiente a esta combinación.

### Subproblema 1

Implementa cualquier estrategia que obedezca las reglas del juego y que termine dentro del límite de tiempo.

Por ejemplo, hay una estrategia simple que siempre hace exactamente  $2^* (49 + 48 + \dots + 2 + 1) = 2450$  llamadas a **faceup (C)**.

### **Subproblema 2**

Implementa una estrategia que termine cualquier juego posible con un máximo de 100 llamadas a **faceup (C)**.