

CI-0112 Programación 1- Grupo 3
Quiz-extra examen (10%)
110 min

110 min

Observaciones

1. Cuenta con 110 minutos para entregar su solución del examen.
2. Como nombre de su proyecto indique su carne junto con ex1, ej: B82345_eq.
3. En el comentario principal de cada clase, como autor indique su número de carné y su nombre.
4. En cada clase, incluya comentarios con el número de pregunta que está respondiendo (marque el inicio y el fin de cada respuesta). Ej:
//inicio respuesta a pregunta #1 15%.
5. El examen es individual. Es prohibido utilizar herramientas digitales para intercambio de documentos. También queda prohibido el acceso de cualquier otro material.
6. Guarde su celular. Si lo utiliza para lo que sea se considerará fraude.

Se tiene un vector que almacena 4 dígitos (entre 1 y 9)

Se le pide implementar usando vectores tanto la generación aleatoria de los 4 valores, como la posibilidad de que el usuario pueda efectuar corrimientos de posición hacia la derecha de estos dígitos. Por ejemplo si el vector (o Ficha) inicial fuera

6	1	5	6
---	---	---	---

Si el usuario indica **desplace (x)** se interpretaría como que deben efectuarse x desplazamientos consecutivos a la derecha (para evitar el desborde, cada vez que un valor se desplaza, el que está a la mas a derecha pasa a la primera posición a la izquierda).

Ejemplo, **desplace (2)** generaría el nuevo estado del vector o Ficha:

5	6	6	1
---	---	---	---

Por otra parte, en el juego se tienen 5 de estos vectores o Fichas. Usted debe acomodar las Fichas de manera tal que el primer y ultimo valor de cada Ficha coincida con las fichas alrededor (la anterior y la siguiente).

entre las funcionalidades que el usuario debe poder indicar se hallan:

- **genere ()** que genera una instanciación que cumple con las reglas (**OJO este método no hace visible los vectores**). Ejemplo con 4 Fichas:

Ficha1

5	6	6	1
---	---	---	---

Ficha2

1	1	3	9
---	---	---	---

Ficha3

9	5	2	2
---	---	---	---

Ficha4

2	2	8	7
---	---	---	---

La situación inicial (y oculta al usuario) debe ser la **solución final** (puede que por casualidad hayan otras situaciones equivalentes que podrían considerarse satisfactorias. Note que el dígito final de cada ficha (exceptuando la última) coincide con el dígito inicial de la siguiente ficha, es decir que el dígito inicial de cada ficha (exceptuando la primera, coincide con el dígito final de la ficha anterior).

- **disperse()** que usando solo la función desplazar(x) y la reasignación de lugar de los vectores, prepara el estado inicial del juego. **OJO este método SI hace visible los vectores.**
Ejemplo con 4 Fichas:

Ficha1

9	1	1	3
---	---	---	---

Ficha2

8	7	2	2
---	---	---	---

Ficha3

6	6	1	5
---	---	---	---

Ficha4

9	5	2	2
---	---	---	---

- **use (i)** que indica el vector con que se va a trabajar
- **desplace(x)** que indica un desplazamiento a la derecha en el vector i de x posiciones
- **asigne (x, y)** que asigna en la posición x al vector y, la ficha y se inserta en la posición x. Las demás fichas bajan a partir de x una posición

Ejemplo: asumiendo la distribución anterior, asigne(2,4), produciría:

Ficha1

9	1	1	3
---	---	---	---

Ficha2

9	5	2	2
---	---	---	---

Ficha3

8	7	2	2
---	---	---	---

Ficha4

6	6	1	5
---	---	---	---

- el método **haGanado()** se debe activar luego de cada movimiento hecho por el usuario.
- automáticamente debe **mostrarse** el estado de cada una de las fichas

Evaluación

15% Diseño de las clases (atributos y métodos definidos)

20% controlador del juego

10% cada uno de los métodos (exceptuando el 3 (use) que vale 5%)

1. genere
2. disperse
3. use
4. displace
5. asigne
6. haGanado
7. muestre