

Ayudantía N°6 – INFO023

- 1) Crear un programa que genere 100 números enteros aleatorios dentro del intervalo [1,1000]. A medida que se generan, se van guardando en un arreglo unidimensional.

Finalmente, el programa imprime en pantalla el número mayor y el menor de todos los que se guardaron en el arreglo (se debe recorrer el arreglo). **[CÓDIGO]**

Se deben usar los siguientes métodos:

- `public static int[] creaArreglo()`: Inicializa el arreglo y lo “rellena” con los números aleatorios generados.
- `public static int numeroMenor(int[] a)` Recibe un arreglo unidimensional de enteros y retorna el número menor de dicho arreglo.
- `public static int numeroMayor(int[] a)` Recibe un arreglo unidimensional de enteros y retorna el número mayor de dicho arreglo.

Ejemplo E/S:

```
E1 numero menor generado fue: 9
E1 numero mayor generado fue:996
```

En los años 80's, una compañía estaba creando un juego y por razones económicas su desarrollo fue cancelado. En la actualidad, un grupo de estudiantes ha obtenido el código original y lo ha reescrito en Java.

Hasta ahora han escrito por completo el método "main", pero no han podido crear los métodos necesarios para su funcionamiento.

[\[Correr código\]](#) [\[Ver código\]](#) [\[Descargar plantilla\]](#)

Las reglas del juego son las siguientes:

- 1) El objetivo del juego es que, dado un tablero (arreglo bidimensional de enteros, cuadrado), cada jugador debe ingresar una coordenada que corresponde a una fila y una columna.
 - 2) Se debe validar que la entrada del usuario tenga el siguiente formato:
 - El largo de la entrada debe ser mayor o igual a 3.
 - La entrada debe indicar fila y columna, separados por una coma (fila,columna).
 - Sólo se permiten caracteres numéricos y la coma.
 - La fila y columna no puede exceder de las dimensiones del arreglo bidimensional.
 - En caso de que la entrada no cumpla con lo anteriormente dicho, se debe solicitar nuevamente.
 - 3) El usuario compite contra el computador.
 - 4) Cada jugador (usuario y computador) tiene 3 vidas.
 - 5) En el tablero, además del tesoro, se encuentran distribuidas cierta cantidad de bombas y botiquines.
 - 6) En caso de ingresar una coordenada donde se encuentre una bomba, se le descontará 1 de vida al jugador correspondiente (para un botiquín, se aumentará 1 de vida).
 - 7) El juego se acaba cuando uno de los jugadores muera o encuentre el tesoro.
 - 8) Valores dentro del tablero:
 - -1: Bomba
 - 0: Tesoro
 - 1: Botiquin
 - 2: Nada
 - 3: Celdas ya registradas por los jugadores
- Completar el programa del juego, escribiendo los métodos faltantes indicados a continuación:
- EL MÉTODO MAIN NO SE DEBE MODIFICAR!!!!.**
- `public static int verCasilla(int [][]tablero,int f,int c):`
recibe el arreglo del tablero de juego y las coordenadas ingresadas, devuelve el número contenido en la casilla correspondiente a dichas coordenadas.
 - `public static int obtenerFila(String s):`
Recibe la entrada ingresada por el usuario (ya validada), extrae y convierte a entero, el número correspondiente a la fila. Retorna ese número.
 - `public static int obtenerColumna(String s):`

- Recibe la entrada ingresada por el usuario (ya validada), extrae y convierte a entero, el número correspondiente a la columna. Retorna ese número.
- `public static boolean analizaRespuesta(String s):`
Recibe la entrada ingresada por el usuario, devuelve true si la entrada es correcta o false si no lo es.
- `public static int[][] creaTablero(int dimension):`
Recibe la dimensión del tablero a crear, crea un arreglo bidimensional de tamaño `dimensionXdimension`. Inicializa todos los elementos del arreglo con el número 2 y lo retorna.
- `public static int[][] colocaBombas(int [][] tablero,int bombas):`
Recibe el tablero de juego (arreglo bidimensional) ya creado y el número de bombas a colocar. Crea una copia del tablero y aleatoriamente, coloca las bombas en ella, asignando el número -1 a las casillas correspondientes y luego retorna la copia del arreglo con las bombas ya colocadas.
- `public static int[][] colocaBotiquines(int [][] tablero,int botiquines)`
Recibe el tablero de juego con las bombas ya puestas y la cantidad de botiquines a distribuir. Crea una copia del tablero.
De forma aleatoria se distribuyen los botiquines en la copia del tablero, asignando el número 2 en las casillas que no tienen bombas. Finalmente se retorna la copia del arreglo.
- `public static int[][] colocaTesoro(int [][] tablero):`
Recibe el tablero de juego (arreglo bidimensional) ya creado con las bombas y botiquines ya colocados. Crea una copia del tablero y aleatoriamente, coloca el tesoro, asignando el valor 0 en alguna de las casillas que no tienen ni bombas ni botiquines.
- `public static void imprimeTablero(int[][] tablero, boolean revelar):`
Recibe el tablero de juego y lo imprime, el booleano es opcional. Si `revelar` es true, se imprime la solución del juego, mostrando donde están las bombas, botiquines y el tesoro, sino, se imprime el tablero y sólo se muestra descubierto las casillas ya registradas por los jugadores.

Algunas imágenes de referencia:

```

Creando tablero...
Listo!
Colocando bombas...
Listo!
Colocando botiquines...
Listo!
Colocando tesoro!!!!...
Listo :D, esta bien escondido ^^
-----
  0  1  2  3  4  5  6  7  8  9
0 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
1 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
2 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
3 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
4 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
5 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
6 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
7 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
8 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]
9 [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

Vidas: Jugador=3 Computador=3
Botiquines:10 Bombas:10

Turno: 1 Juega: [Jugador]

Ingrese celda a inspeccionar (fila,columna)
>: _

```

```

Turno: 1 Juega: [Jugador]

Ingrese celda a inspeccionar (fila,columna)
>: 3
Ingrese celda a inspeccionar (fila,columna)
>: 33,
Ingrese celda a inspeccionar (fila,columna)
>: 2,
Ingrese celda a inspeccionar (fila,columna)
>: 5,,0
Ingrese celda a inspeccionar (fila,columna)
>: 5,0
Ingrese celda a inspeccionar (fila,columna)
>: fila ingresada: 5
columna ingresada: 0
No se ha encontrado nada

```

```

Solucion:
  0  1  2  3  4  5  6  7  8  9
0 [ ] [ ] [ ] [B] [ ] [ ] [ ] [ ] [ ] [ ]
1 [ ] [ ] [ ] [ ] [B] [ ] [ ] [ ] [ ] [ ]
2 [ ] [0] [ ] [ ] [ ] [ ] [B] [ ] [ ] [ ]
3 [B] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
4 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [B] [ ] [ ]
5 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [B] [ ] [ ]
6 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
7 [ ] [ ] [ ] [ ] [B] [ ] [ ] [ ] [ ] [ ]
8 [ ] [B] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [B]
9 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
Ingrese celda a inspeccionar (fila,columna)

```