

0485.- Programación.

U3 - Entregable_1920

Ejercicio 1

Un restaurante nos ha encargado una aplicación para colocar a los clientes en sus mesas. En una mesa se pueden sentar de 0 (mesa vacía) a 4 comensales (mesa llena). Cuando llega un cliente se le pregunta cuántos son. De momento el programa no está preparado para colocar a grupos mayores a 4, por tanto, si un cliente dice por ejemplo que son un grupo de 6, el programa dará el mensaje "Lo siento, no admitimos grupos de 6, haga grupos de 4 personas como máximo e intente de nuevo". Para el grupo que llega, se busca siempre la primera mesa libre (con 0 personas). Si no quedan mesas libres, se busca donde haya un hueco para todo el grupo, por ejemplo si el grupo es de dos personas, se podrá colocar donde haya una o dos personas. Inicialmente, las mesas se cargan con valores aleatorios entre 0 y 4. Cada vez que se sientan nuevos clientes se debe mostrar el estado de las mesas. Los grupos no se pueden romper aunque haya huecos sueltos suficientes. El funcionamiento del programa se ilustra a continuación:

Ejemplo:

Mesa no	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	0	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 2
Por favor, siéntense en la mesa número 3.

Mesa no	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4
Por favor, siéntense en la mesa número 7.

Mesa no	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 3
Tendrán que compartir mesa. Por favor, siéntense en la mesa número 6.

Mesa no	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4



Lo siento, en estos momentos no queda sitio.

Mesa no	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): -1
Gracias. Hasta pronto.

Ejercicio 2

Escribe un programa que pida 20 números enteros. Estos números se deben introducir en un array de 4 filas por 5 columnas. El programa mostrará las sumas parciales de filas y columnas igual que si de una hoja de cálculo se tratara. La suma total debe aparecer en la esquina inferior derecha.

					Σ fila 0
					Σ fila 0
					Σ fila 0
					Σ fila 0
Σ columna 0	Σ columna 1	Σ columna 2	Σ columna 3	Σ columna 4	TOTAL

Ejercicio 3

Crea la función de manejo de [arrays](#) que tenga la siguiente cabecera y que haga lo que se especifica en los comentarios:

```
public int[] filtraCon7(int x[]) // Devuelve un array con todos los números
                                // que contienen el 7 (por ej. 7, 27, 782)
                                // que se encuentren en otro array que se
                                // pasa como parámetro. El tamaño del array
                                // que se devuelve será menor o igual al
                                // que se pasa como parámetro.
```

Utiliza esta función en un programa para comprobar que funcionan bien. Para que el ejercicio resulte más fácil, las repeticiones de números que contienen 7 se conservan; es decir, si en el array x el número 875 se repite 3 veces, en el array devuelto también estará repetido 3 veces. Si no existe ningún número que contiene 7 en el array x, se devuelve un array con el número -1 como único elemento.

Ejercicio 4

Define la función convierteArrayEnString con la siguiente cabecera:

```
public static String convierteArrayEnString(int[] a)
Esta función toma como parámetro un array que contiene números y devuelve una cadena de caracteres con esos números. Por ejemplo, si a = { }, convierteArrayEnString(a) devuelve ""; si a = { 8 }, convierteArrayEnString(a) devuelve "8"; si a = { 6, 2, 5, 0, 1 }, convierteArrayEnString(a) devuelve "62501".
```

Ejercicio 5



Implementa la función aleatorioDeArray con la cabecera que se muestra a continuación:

```
public static int aleatorioDeArray(int[] a)
```

Esta función debe devolver un número del array escogido al azar entre todos los disponibles. Por ejemplo, si `a = {111, 222, 333, 444}`, `aleatorioDeArray(a)` podría devolver el 111, el 222, el 333 o el 444. Si `b = {52, 37}`, `aleatorioDeArray(b)` podría devolver el 52 o el 37. Utiliza la función en un programa de prueba.

Estado de la entrega

Estado de la entrega	No entregado		
Estado de la calificación	Sin calificar		
Criterio de calificación	Rúbrica Entrega U3		
Ej1 - Lectura de dato	No se entrega o no se hace correctamente <i>0 puntos</i>	Solicita el número de comensales pero o no comprueba su número o no lo hace iterativamente o no puede salir con -1 <i>1 puntos</i>	Solicita el número de comensales, comprueba su número y lo hace iterativamente, además de poder salir con -1. <i>3 puntos</i>
Ej1 - Inicio array del restaurante	No se entrega o no se hace correctamente <i>0 puntos</i>		Rellena el array inicial del restaurante con una ocupación aleatoria. <i>3 puntos</i>
Ej1 - Bucle 1. Mesas vacías	No se entrega o no lo establece correctamente <i>0 puntos</i>		Establece el bucle para comprobar que hay alguna mesa completamente vacía <i>8 puntos</i>
Ej1 - Bucle 2. Mesas en las que cabe.	No se entrega o no establece correctamente la salida. <i>0 puntos</i>		Establece el bucle para comprobar que hay alguna mesa en la que se cabe sin partir el grupo. <i>10 puntos</i>
Ej 1 - Actualización del nivel de ocupación	No se entrega o no actualiza correctamente la ocupación. <i>0 puntos</i>	Actualiza correctamente la ocupación del restaurante pero no imprime el estado cada vez. <i>2 puntos</i>	Actualiza correctamente la ocupación del restaurante e imprime el estado cada vez. <i>5 puntos</i>
Ej2 - Lectura de datos desde usuario	No se entrega o no se hace correctamente <i>0 puntos</i>		Muestra el mensaje solicitando la entrada de números. <i>5 puntos</i>

