PRUEBA PROGRAMADOR **JAVA**

Nombre: Fecha:

1. Resuelva el siguiente problema creando una aplicación Java:  
     
   Un barrio de ocho casas, representadas como celdas, están organizadas en línea recta. Cada día, todas las celdas compiten con sus celdas adyacentes (vecinos). Un valor entero de 1 representa una celda activa y un valor 0 representa una celda inactiva. Sí ambos vecinos están activos o inactivos, la celda cambia su estado a inactivo el siguiente día. De lo contrario, la celda cambia su estado a activo el siguiente día.  
   Las celdas de inicio y final al tener solamente una celda adyacente se puede asumir que el vecino faltante siempre tendrá un estado inactivo. Se debe tener en cuenta que incluso después de actualizar el valor de la celda en proceso, su valor anterior es considerado para actualizar el valor de las otras celdas.  
     
   Defina un algoritmo que muestre el estado de las celdas después de un número de días determinado.  
     
   **Entradas:**- Casas: Lista de enteros que representa el estado actual de las celdas. Este elemento solo tendrá valores 1 y 0.  
   - Días: Un entero que representa la cantidad de días.  
   **Salidas:**Retorne un listado de enteros que muestre el estado de las celdas después de un determinado número de días.  
    **Ejemplos de Salida**  
   Cantidad de días: 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Salida | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

Cantidad de días: 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Salida | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

1. Resuelva el siguiente problema creando una aplicación en el lenguaje que desee:

En un centro de entrega de una compañía de envío de paquetes, una computadora decide cuales paquetes serán cargados en los camiones de la compañía.

Todos los espacios en los camiones son clasificados como **unidades de tamaño**, los cuales son representados con números enteros. Cada tipo de camión, tiene sus propias unidades de tamaño. Así mismo, cada paquete ocupará una cantidad de unidades de tamaño específica.

Teniendo en cuenta un listado de paquetes y el tamaño del camión (parámetros de entrada), se debe crear un programa que se encargue de determinar, exactamente, una pareja de paquetes que se puedan cargar en el camión.

**Reglas de negocio:**

* 1. El camión **debe reservar 30 unidades de tamaño siempre**, por seguridad.
  2. Se cargarán exactamente dos paquetes por camión, cuyo tamaño sumado permita dejar exactamente 30 unidades de tamaño libres en el camión.
  3. No se puede escoger el mismo paquete dos veces.
  4. Si existe la posibilidad de cargar dos pares de paquetes en el camión, se debe escoger el par que tenga el paquete más grande.

**Parámetros de entrada:**

1. **TamanioCamion:** Un entero que representa el tamaño del camión.
2. **Paquetes:** Listado de enteros que representa el número de unidades de tamaño que ocupará cada paquete en el camión.

**Salida/Retorno:**

Listado de enteros que contiene los dos paquetes seleccionados para ser cargados en el camión.

**Ejemplo:**

Tamaño del Camión: 90

Paquetes: [10, 60, **40**, 35, **20**]

Resultado: [40,20] -> La suma de los paquetes es 60, lo que permite dejar las 30 unidades de espacio requeridas.

1. En este diagrama por capas, especifique los componentes separados con comas que conoce y dispone el estándar de arquitectura JEE7

|  |  |
| --- | --- |
| **Capa** | **Componentes JEE7** |
| Presentación | Java Services Faces |
| Servicios | Servicios con SpringBoot, Servicios Rest |
| Datos | JPA - Hibernate |