Dada una matriz de caracteres de MxN compuesta de secuencias de caracteres separadas por espacios, hacer un programa completo para borrar la secuencia central de cada fila (secuencia correspondiente a la mitad de la cantidad de secuencias que tiene la fila, sin importar si la cantidad es par o impar). Para borrar realizar los desplazamientos necesarios y no utilizar estructuras auxiliares. Suponer que la matriz se encuentra cargada, y que cada fila comienza y termina con caracteres espacio.

```
public class SegundoParcial2019 {
	final static int M = 5;
	final static int N = 15;
	public static void main(String[] args) {
		char[][] mat = {{' ','1','2','1',' ','3','4',' ',' ','1','2',' ','1',' ',' '},
				{' ','3','4','5',' ','1','2',' ',' ','1','1',' ',' ',' ',' '},
				{' ',' ',' ',' ',' ',' ',' ',' ',' ','1','1',' ',' ',' ',' '},
				{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
				{' ',' ',' ',' ',' ','3','4',' ',' ','1','1',' ','1',' ',' '}};
		imprimir_matriz(mat);
		eliminar_secuencias_centrales(mat);
		imprimir_matriz(mat);
	}

	public static void eliminar_secuencias_centrales(char[][] mat){
		int inicio = 0;
		int fin = -1;
		int cant = 0;
		for (int fil = 0;fil < M;fil++){
			cant = cantidad_secuencias_fila(mat,fil);
			if (cant > 0){
				inicio = obtener_inicio_secuencia(mat,fil,cant/2+1);
				fin = obtener_fin_secuencia(mat,fil,inicio);
				for(cant=fin-inicio+1;cant>0;cant--)
					corrimiento_izq_fila(mat,fil,inicio);
			}
		}
	}

	public static void corrimiento_izq_fila(char[][] mat, int fil, int pos){
		for (int col=pos;col<N-1;col++)
			mat[fil][col]=mat[fil][col+1];
	}

	public static int cantidad_secuencias_fila(char[][] mat, int fil){
		int cant = 0;
		for (int col=0;col<N-1;col++){
			if ((mat[fil][col]==' ')&&(mat[fil][col+1]!=' '))
				cant++;
		}
		return cant;
	}

	public static int obtener_inicio_secuencia(char[][] mat, int fil, int cant){
		int inicio = 0;
		while ((inicio<N-1)&&(cant>0)){
			if ((mat[fil][inicio]==' ')&&(mat[fil][inicio+1]!=' '))
				cant--;
			inicio++;
		}
		return inicio;
	}

	public static int obtener_fin_secuencia(char[][] mat, int fil, int inicio){
		while ((inicio < N)&&(mat[fil][inicio]!=' '))
			inicio++;
		return inicio-1;
	}
```

```java
	public static void imprimir_matriz(char[][] mat) {
		System.out.println("--------------------");
		for (int i = 0 ; i < M; i++){
			for (int j = 0 ; j < N; j++)
				System.out.print(" "+mat[i][j]);
			System.out.println("");
		}
	}
}
```