Enunciado:

Resuelva los siguientes ejercicios sobre implementación de estructuras de datos secuenciales o lineales: `stack` y `queue`.

Utilice el estándar C++14 en la solución de sus problemas. No olvide compilar con los *flags* apropiados para detectar *warnings* y errores.

1. Implemente una versión sencilla del contenedor secuencial `std::stack`, llamada `Stack`, usando el concepto de listas enlazadas (*linked lists*). Los métodos y variables de instancia que deben ser implementados se muestran a continuación.

```
template <typename dataType>
class Stack {
private:
    struct Cell {       /* Type for linked list cell */
        dataType data;
        Cell *link;
    };

    Cell *stack;        /* Beginning of the list */
    int count;          /* Number of elements */

    void deepCopy(const Stack<dataType> & src);

public:
    Stack();
    ~Stack();
    size_t size();
    bool empty();
    void clear();
    void push(dataType & ch);
    dataType pop();
    dataType peek();
};
```

   Si considera que más métodos son esenciales en su implementación, no dude en incluirlos.

2. Implemente una versión simplificada del contenedor secuencial `std::queue`, llamada `Queue`, usando arreglos dinámicos (*dynamic arrays*). Los métodos y variables de instancia que deben ser implementados se muestran a continuación.

```
template <typename dataType>
class Queue {
private:
    static const int INITIAL_CAPACITY = 10;

    dataType *array;
    int capacity, head, tail;
```

```
8
9       void expandCapacity();
10      void deepCopy(const Queue<dataType> & src);
11
12  public:
13      Queue();
14      ~Queue();
15      int size();
16      bool empty();
17      void clear();
18      void enqueue(dataType & ch);
19      dataType dequeue();
20      dataType peek();
21  };
```

Si considera que más métodos son esenciales en su implementación, no dude en incluirlos.