

Ejercicios Estructuras de Datos Lineales

1. Dada una cola de enteros y una lista diremos que son isomórficas si contienen los mismo elementos en las posiciones pares. Construir la siguiente función

bool isomorficas (queue<int> & C, list<int> & L);

Ejemplo: (La primera posición es la 1)

C= {1,5,7,9,8,4} y L={9,5,3,9,10,4} devolvería True

^

front

C= {1,5,7,9,8,1} y L={9,5,3,9,10,4} devolvería False

^

Front

2. Dada una cola de listas de enteros devolver otra cola de enteros que contenga en cada posición la suma de los elementos en la lista correspondiente en la cola original. La cabecera de la función sería la siguiente:

queue<int> GetSumas(queue<list<int> > & ql);

Ejemplo

ql={{1,2,3},{4,5},{3,7,2}} → Qout={6,9,12}

^

front

3. Dada una lista de enteros y una pila de enteros deducir si todos los elementos de la lista están contenidos en la pila y además contienen el mismo numero de enteros. La cabecera de la función es:

bool es_permutacion(const list<int> &L, const stack<int> & q);

Ejemplo:

L={1,2,4,5} y q={5,4,1,2} devolvería True

L={1,3,9} y q={3,9,9,1} devolvería False

4. Definir la clase MultiPilas como una lista de pilas. Definir los siguientes métodos:
 - Multitop: obtiene una lista con los topes de todas las pilas
 - Multipop: elimina los topes de todas las listas
 - Multipush: dada una lista de enteros añade esos elementos a las pilas. Así el elemento primero de la lista de entrada se le añade a la pila primera, el segundo a la segunda pila y así sucesivamente.
 - top(list<stack<int> >::iterator it): obtiene el tope de la pila apuntada por el iterador
 - pop(list<stack<int> >::iterator it) elimina el tope de la pila apuntada por el iterador
 - push(list<stack<int> >::iterator, int v): pone el element v en la pila aputada por el iterador.
 - size: devuelve el número de pilas

- `size(list<stack<int>>::iterator)` devuelve el numero de elementos de la pila apuntada por el iterador.
- Insert: inserta una nueva pila por el final.
- Erase: borra la pila indicada por un iterador