

EJERCICIOS TAD LINEALES

Realizar los siguientes ejercicios usando WxDevC++.

1. Definir el Tipo de dato **Lista** de números enteros con gestión dinámica. La lista debe ser ordenada y debe contener un nodo cabecera vacío y terminar siempre con un nodo NULL.

Habrás que crear las siguientes funciones:

Lista CreateList(void): debe devolver una lista creada que conste de nodo cabecera vacío y nodo final NULL.

void ViewList(Lista): debe mostrar los elementos de una lista proporcionada.

void InsertElementList(Lista,v): debe insertar el elemento con valor v en el orden adecuado.

Int CountElementList(Lista,v): debe devolver un número indicando el número de veces que aparece un valor en la lista.

Lista FindElementList(Lista,v): debe devolver un puntero al nodo cuyo valor es v, si existe y sino un puntero a NULL.

int LongList(Lista): debe devolver el número de elementos de la lista.

bool EmptyList(Lista): debe devolver un booleano indicando si la lista está vacía o no.

void EraseElementList(Lista,v): debe borrar un elemento en la lista, si existe.

2. Definir el Tipo de dato **Cola** de números enteros con gestión dinámica. La cola debe contener un nodo cabecera vacío y debe terminar siempre con un nodo NULL.

Habrás que crear las siguientes funciones:

Cola CreateQueue(void): debe devolver una cola creada que conste de nodo cabecera vacío y nodo final NULL.

void ViewQueue(Cola): debe mostrar los elementos de una cola en el orden adecuado.

void InQueue(Cola,v): debe insertar el valor v en el orden de llegada que le corresponda sin tener en cuenta su valor.

int OutQueue(Cola): debe devolver el elemento siguiente a procesar eliminándolo de la cola.

int FirstQueue(Cola): debe devolver el primer elemento de la cola a procesar sin sacarlo de la misma.

int LastQueue(Cola): debe devolver el último elemento de la cola a procesar sin sacarlo de la misma.

int LongQueue(Cola): debe devolver el número de elementos de la cola.

bool EmptyQueue(Cola): debe devolver un booleano indicando si la cola está vacía o no.

void InRandQueue(Cola,n): debe insertar n elementos numéricos generados al azar dentro de la cola.

3. Definir el Tipo de dato **Pila** de números enteros con gestión dinámica. La pila debe contener un nodo cabecera vacío y debe terminar siempre con un nodo NULL.

Habría que crear las siguientes funciones:

Pila CreateStack(void): debe devolver una pila creada que conste de nodo cabecera vacío y nodo final NULL.

void ViewStack(Pila): debe mostrar los elementos de una pila proporcionada.

void PushStack(Pila,v): debe apilar el elemento con valor v.

int PopStack(Pila): debe desapilar el último elemento de la pila y devolverlo (suponiendo que sea un entero). No se podrá desapilar de una pila vacía.

int TopStack(Pila): debe devolver el valor del último elemento de la pila sin desapilarlo.

int HighStack(Pila): debe devolver el número de elementos de la pila.

bool EmptyStack(Pila): debe devolver un booleano indicando si la pila está vacía o no.

void PushRandStack(Pila,n): debe apilar n elementos numéricos generados al azar dentro de la pila.