

Tipos de datos lineales.

ED Grupo A. Profesor: Isabel Pita.

Nombre del alumno:

1. Para obtener un coste constante en todas las operaciones de una pila (push, pop, top, empty) la cima de la pila debe situarse:

Si la implementación se realiza con un array:

Si la implementación se realiza con una lista enlazada simple:

2. A una estación de tren con una vía muerta, llega un tren con 5 vagones: 1,2,3,4,5. El jefe de estación debe reordenar los vagones de forma que al salir de la estación el tren tenga los vagones en el orden: 4,5,3,2,1. El maquinista solo puede llevar un vagón a la vía muerta, sacar un vagón de la vía muerta, o llevar un vagón directamente de la entrada de la estación a la salida (sin pasar por la vía muerta). Indica si es posible que se forme el tren pedido.

Escribe otros 2 posibles trenes que puedan salir de la estación y 2 que no puedan formarse.

¿Es posible? SI NO.

Si contestas SI, indica como lo hace:

Trenes que se pueden formar:

Trenes que no se pueden formar:

3. Queremos implementar una cola y para ello sólo contamos con el TAD pila (no existen vectores ni listas enlazadas) ¿Cuál es el menor número de pilas necesario para implementarla?. Escribe el código de la función pop de la cola implementada de esta manera.

4. ¿Qué hace la siguiente función cuando la llamamos con el primer nodo de una lista enlazada?

```
void fun(Nodo * prim) {  
    if (prim != nullptr) {  
        fun(prim->sig);  
        cout << prim->elem << ' ';  
    }  
}
```

- a) Imprime todos los nodos de la lista
 - b) Imprime todos los nodos de la lista en orden inverso
 - c) Imprime nodos alternos de la lista
 - d) Imprime nodos alternos de la lista en orden inverso
5. Completa el siguiente código que mueve el último elemento de una lista enlazada simple al principio

```
void adelanta(Nodo * & prim) {  
    if (prim == nullptr || prim->sig == nullptr) return;  
    Nodo * p = prim, * q = nullptr;  
    while (p->sig != nullptr) {  
        q = p; p = p->sig;  
    }  
    // instrucciones que faltan  
  
}
```

6. ¿Cuántas comparaciones son necesarias en el caso peor para buscar un elemento en una lista enlazada de longitud n ? Justifica la respuesta

7. ¿Qué operación es más eficiente cuando se utilizan listas doblemente enlazadas con punteros al comienzo y al final de la lista frente a utilizar una lista enlazada simple con punteros al comienzo y al final de la lista?

- a) Añadir un elemento por el principio de la lista
- b) Añadir un elemento por el final de la lista
- c) Borrar un elemento por el principio de la lista
- d) Borrar un elemento por el final de la lista.

8. Escribe las instrucciones que eliminan el nodo apuntado por un puntero X de una lista doblemente enlazada?

```
struct Nodo {  
    T elem;  
    Nodo * ant;  
    Nodo * sig;  
};
```