

Práctica 1. Estructuras de datos avanzadas

P1.1. En este ejercicio debes completar la librería para trabajar con pilas de enteros. Tienes a tu disposición el fichero `pilaEnterosDinamica.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `pilaEnterosDinamica.c`, con la implementación de todas las funciones exportadas. Ten en cuenta que la pila debe ser dinámica, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en ella. Las funciones que debes implementar son:

- Crear una nueva pila
- Apilar un elemento
- Desapilar un elemento
- Devolver el elemento de la cima
- Comprobar si la pila está vacía

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo `ejecutarPilaEnteros.c`, que contiene un programa con la función `main`. Este fichero no debes modificarlo.

Ejemplo:

```
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 1
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 3
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 3
La cima es 3
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 2
Se ha eliminado un elemento
```

```
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 3
La cima es 1
-----MENU-----
1 - Apilar un elemento
2 - Desapilar un elemento
3 - Mostrar elemento de la cima.
4 - Salir.
Escoja una opcion: 4
```

P1.2. En este ejercicio debes completar la librería para trabajar con colas de enteros. Tienes a tu disposición el fichero `colaEnteros.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `colaEnteros.c`, con la implementación de todas las funciones exportadas. Ten en cuenta que la cola debe ser dinámica, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en ella. Las funciones que debes implementar son:

- Crear una nueva cola
- Encolar un elemento
- Desencolar un elemento
- Devolver el elemento del frente
- Comprobar si la cola está vacía

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo `ejecutarColaEnteros.c`, que contiene un programa con la función `main`. Este fichero no debes modificarlo.

Ejemplo:

```
-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 1
-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 3
-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 3
El frente es 1

-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 2
Se ha eliminado un elemento
-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 3
El frente es 3
-----MENU-----
1 - Encolar un elemento
2 - Desencolar un elemento
3 - Mostrar elemento del frente.
4 - Salir.
Escoja una opcion: 4
```

P1.3. Implementa en un archivo `.c` la solución al siguiente problema. Para la resolución, puedes utilizar las librerías que has creado en los ejercicios anteriores.

Vamos a trabajar con un vector de registros. Cada registro consta de tres campos con información sobre grandes desastres:

- Fecha del desastre
- Número de víctimas
- La fecha del último desastre anterior que tuvo más víctimas que el presente

Pide información al usuario para rellenar los dos primeros campos de todos los registros del vector. Esta información se va a introducir ordenada cronológicamente, desde el desastre más antiguo al más reciente. En todos los registros, el tercer campo está vacío.

El objetivo del programa es completar el vector, rellenando el tercer campo de cada registro a partir de la información que ya contiene el vector. Si no hay ningún desastre anterior con más víctimas que el actual, este campo se debe rellenar a 0.

Escribe un algoritmo que utilice una estructura auxiliar (pila o cola, la que consideres que es más adecuada) y tenga un coste en tiempo de $O(N)$, es decir, que solo recorra los elementos del vector una única vez.

Ejemplo:

```
Introduce el numero de desastres: 6
Introduce la fecha y el numero de victimas: 19900512 150000
Introduce la fecha y el numero de victimas: 19950131 300000
Introduce la fecha y el numero de victimas: 20011203 20000
Introduce la fecha y el numero de victimas: 20020606 140000
Introduce la fecha y el numero de victimas: 20070322 60000
Introduce la fecha y el numero de victimas: 20130728 250000
19900512 150000 0
19950131 300000 0
20011203 20000 19950131
20020606 140000 19950131
20070322 60000 20020606
20130728 250000 19950131
```

P1.4. En este ejercicio debes completar la librería para trabajar con secuencias de enteros. Una secuencia es una estructura lineal con un punto de interés donde se realizan las modificaciones y las consultas. Cada secuencia se debe implementar utilizando dos pilas dinámicas: una almacena los números enteros hasta el anterior al punto de interés, la otra almacena los números enteros desde el punto de interés hasta el final.

Tienes a tu disposición el fichero `secuenciaEnteros.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `secuenciaEnteros.c`, con la implementación de todas las funciones exportadas. Las funciones que debes implementar son:

- Crear una secuencia vacía
- Insertar un elemento delante del punto de interés
- Insertar un elemento en el punto de interés
- Eliminar el elemento del punto de interés
- Consultar el elemento del punto de interés
- Avanzar en una posición el punto de interés
- Trasladar el punto de interés al comienzo de la secuencia
- Determinar si el punto de interés está al final de la secuencia
- Comprobar si la secuencia está vacía

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo `ejecutarSecuenciaEnteros.c`, que contiene un programa con la función `main`. Este fichero no debes modificarlo.

Ejemplo:

```

1-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 1
Introduce el entero: 5
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 1
Introduce el entero: 8
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 2
Introduce el entero: 4

```

```

-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 4
El elemento en el punto de interes es 4
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 5
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 7
El punto de interes esta en el ultimo elemento de la secuencia

```

```

-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 6
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 4
El elemento en el punto de interes es 5
-----MENU-----
1 - Insertar un elemento delante del punto de interes
2 - Insertar un elemento en el punto de interes
3 - Eliminar un elemento del punto de interes
4 - Consultar el elemento en el punto de interes.
5 - Avanzar un elemento el punto de interes
6 - Mover el punto de interes al principio
7 - Esta el punto al final?
8 - Esta vacia la secuencia?
9 - Salir
Escoja una opcion: 9

```

P1.5. Implementa en un archivo .c la solución al siguiente problema. Para la resolución, puedes utilizar las librerías que has creado en los ejercicios anteriores, o crear unas muy similares si necesitas que trabajen con otro tipo de datos.

Dado un string que puede contener cualquier carácter estándar, comprobar si la sucesión de paréntesis, llaves y corchetes que aparecen está equilibrada, es decir, si a cada paréntesis de apertura (o llave o corchete) le corresponde uno de su misma categoría pero de cierre, estando bien anidados. Por ejemplo, (a(a){jj}o) está bien equilibrada, pero [a[f(j)]p{g}o] no lo está.

Ejemplo:

```
Introduce la frase: (Esta {es {una} fr(ase) de [[pru]eba])  
Los parentesis no estan balanceados
```

```
Introduce la frase: (Esta {es {una} fr(ase) de [[pru]eba]})  
Los parentesis si estan balanceados
```

P1.6. En este ejercicio debes completar la librería para trabajar con colas doblemente enlazadas de caracteres. Es decir, desde cada carácter de la cola se puede acceder al anterior y al siguiente. Tienes a tu disposición el fichero colaDobleCaracteres.h, que contiene las exportaciones de la librería. Tú debes crear el fichero colaDobleCaracteres.c, con la implementación de todas las funciones exportadas. Ten en cuenta que la cola debe ser dinámica, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en ella. Las funciones que debes implementar son:

- Crear una cola doble vacía
- Encolar por adelante
- Encolar por atrás
- Desencolar por adelante
- Desencolar por atrás
- Mostrar el primer elemento de la cola
- Mostrar el último elemento de la cola
- Comprobar si la cola doble está vacía

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo ejecutarColaDobleCaracteres.c, que contiene un programa con la función main. Este fichero no debes modificarlo.

Ejemplo:

```

-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 1
Introduce el entero: 1
-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 2
Introduce el entero: 5
-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 2
Introduce el entero: 3
-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 5
El primer elemento es 1

```

```

-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 4
Se ha eliminado un elemento
-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 6
El ultimo elemento es 5
-----MENU-----
1 - Encolar un elemento al inicio
2 - Encolar un elemento al final
3 - Desencolar un elemento del inicio
4 - Desencolar un elemento del final
5 - Mostrar el elemento del inicio.
6 - Mostrar el elemento del final.
7 - Salir.
Escoja una opcion: 7

```

P1.7. Implementa en un archivo .c la solución al siguiente problema. Para la resolución, puedes utilizar las librerías que has creado en los ejercicios anteriores, o crear unas muy similares si necesitas que trabajen con otro tipo de datos.

El agente 0069 ha inventado un nuevo método de codificación de mensajes secretos. El mensaje original X se codifica en dos etapas: en primer lugar, X se transforma en X' reemplazando cada sucesión de caracteres consecutivos que no sean vocales por su inversa. En segundo lugar, X' se transforma en la sucesión de caracteres X'' obtenida al ir tomando sucesivamente el primer carácter de X', luego el último, luego el segundo, luego el penúltimo, etc. Por ejemplo,

X = Anacleto, agente secreto

X' = Analceto ,agentes erceto

X'' = Aontaelcreet os e,natge

Implementar el problema para codificar y decodificar los mensajes.

Ejemplo:

```
Introduce la frase a codificar: Anacleto, agente secreto
Tras la primera codificacion el mensaje es: Analceto ,agetnes erceto
Tras la segunda codificacion el mensaje es: Aontaelccreet os e,natge
Tras la primera decodificacion el mensaje es: Analceto ,agetnes erceto
El mensaje decodificado es: Anacleto, agente secreto
```

P1.8. Implementa en un archivo .c la solución al siguiente problema.

En la revuelta judía contra Roma, Josephus y 39 de sus compañeros se resistieron a los romanos en una cueva. Con la inminencia de la derrota, decidieron que preferirían morir antes que ser esclavos de los romanos. Decidieron organizarse en un círculo. Un hombre fue designado como el número uno, y procedieron en sentido horario matando a cada séptimo hombre (paso). Josephus fue entre otras cosas un consumado matemático; por lo que instantáneamente descubrió dónde debería sentarse para ser el último en irse. Pero cuando llegó el momento, en lugar de suicidarse, se unió al bando romano.

Implementar un método que averigüe en qué posición se debe sentar Josephus para no ser asesinado. La solución debe generalizarse para cualquier cantidad de soldados judíos y cualquier paso. En la solución se debe usar una cola de enteros (cada soldado se representa con un número del 1 al n).

Ejemplo:

```
Introduce el numero de resistentes: 11
Introduce el paso (cada cuantos se muere uno): 3
El resistente que no muere es el numero 7
```

P1.9. En este ejercicio debes completar la librería para trabajar con listas ordenadas de enteros doblemente enlazadas. Tienes a tu disposición el fichero listaOrdenadaEnteros.h, que contiene las exportaciones de la librería. Tú debes crear el fichero listaOrdenadaEnteros.c, con la implementación de todas las funciones exportadas. Ten en cuenta que la lista debe ser dinámica, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en ella. Las funciones que debes implementar son:

- Crear una nueva lista
- Insertar un elemento en la lista
- Eliminar el menor de la lista
- Eliminar el mayor de la lista
- Consultar el menor elemento
- Consultar el mayor elemento
- Comprobar si un número está en la lista
- Comprobar si la lista está vacía
- Dadas dos listas, concatenar todos los elementos en la primera lista y vaciar la segunda. Los elementos deben estar completamente ordenados en la primera lista
- Imprimir todos los elementos de la lista

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo ejecutarListaOrdenadaEnteros.c, que contiene un programa con la función main. Este fichero no debes modificarlo.

Ejemplo:

```
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 1
Introduce el entero: 3
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 1
Introduce el entero: 2
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 2
Introduce el entero: 1
```

```
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 8
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 5
1 2 3
-----MENU-----
1 - Insertar un elemento en la primera lista
2 - Insertar un elemento en la segunda lista
3 - Eliminar el menor elemento de la lista 1
4 - Eliminar el mayor elemento de la lista 1
5 - Mostrar todos los elementos de la lista 1.
6 - Mostrar todos los elementos de la lista 2.
7 - Comprobar si un elemento esta en la lista 1.
8 - Concatenar ambas listas.
9 - Salir.
Escoja una opcion: 9
```