

# Ejercicios

# Contenidos

- 1 Ejercicios
  - Listas
  - Pilas
  - Colas
- 2 Mas ejercicios
- 3 Ejercicio 1
- 4 Ejercicio 2
- 5 Ejercicio 3
- 6 Ejercicio 4
- 7 Ejercicio 5
- 8 Ejercicio 6
- 9 Ejercicio 7

# Ejercicios

## Listas

Implementar las siguientes operaciones sobre listas

- Insertar en la  $i$ -ésima posición.
- Eliminar el elemento de la  $i$ -ésima posición.

# Ejercicios

## Pilas

Implementar las siguientes operaciones sobre pilas:

- Insertar en la  $i$ -ésima posición.
- Eliminar el elemento de la  $i$ -ésima posición.

# Ejercicios

## Colas

Implementar las siguientes operaciones sobre colas:

- Insertar en la  $i$ -ésima posición.
- Eliminar el elemento de la  $i$ -ésima posición.

# Ejercicio 1

Se tiene una palabra compuesta solo de  $A$ s y de  $B$ s. Mediante el uso de pilas verifique que un string  $S$  tiene la misma cantidad de  $A$  que de  $B$ .

## Ejercicio 2

Mediante el uso de pilas verifique si el número  $n$  es par.

## Ejercicio 3

Se tiene un entero  $n$ , descomponga el número en sus factores primos y guárdelos en una lista. No debe repetir ninguno de los elementos, pero debe mantener la equivalencia.



## Ejercicio 4

Implementar un procedimiento para insertar un dato en orden ascendente en una lista enlazada. Es decir, el nodo que representa el dato debe ser insertado en una posición tal que al recorrer la lista los nodos se recorran de menor a mayor respecto del dato. Los elementos no se pueden repetir.

# Ejercicio 5

Obtenga la ecuación de recurrencia y su solución para la función `funcionDesconocida( $n$ )`. La función `hermosaYdesconocida( $n$ )` toma tiempo constante  $a$  y la función `hermosa( $n$ )` toma tiempo  $n^2$ . Suponga que  $n$  es potencia 2.

```
1 void funcionDesconocida(int *a, int izq, int der)
2 {
3     int med = (izq + der) / 2;
4     if(izq >= der)
5         hermosaYdesconocida(a, izq, der);
6     else
7     {
8         hermosa(a, izq, der);
9         /* revisar subconjunto de la izquierda. */
10        funcionDesconocida(a, izq, med);
11        /* revisar en el subconjunto de la derecha. */
12        funcionDesconocida(a, med+1, der);
13    }
14 }
```

# Ejercicio 6

Una variante del conocido problema de Josefo es el siguiente: determinar los datos que van quedando de una lista inicial de  $n$  números sometida al siguiente algoritmo:

- i) Se genera un número aleatorio  $s$ .
- ii) Si  $s > n$  se detiene.
- iii) Si  $s \leq n$  se quitan de la lista los números que ocupan las posiciones  $1 + 0s, 1 + 1s, 1 + 2s, \dots$ ;  $n$  toma el número de elementos que quedan en la lista.
- iv) Se vuelve al paso i)

# Ejercicio 7

Realizar la función `unirListas`(Lista \*primera, Lista \*segunda) en C. Esta función debe generar la unión combinando dos listas. Es decir, tomando el primer elemento de la primera, el primero de la segunda, el segundo de la primera, el segundo de la segunda y así sucesivamente, es decir, intercalando los nodos. Debe guardarse en la primera lista (la lista a la que se aplica la función). Si tienen distinto tamaño, se completará la inserción sólo con la lista que le quedan elementos.

# Ejercicios