



UNIVERSIDAD CATÓLICA ANDRÉS BELLO – EXTENSIÓN GUAYANA

Escuela de Ingeniería Informática

Manual de Prácticas: Algoritmos y Estructuras de Datos

Actividad 5

Pilas y Colas

Ejercicio 1

Utilizando solamente las operaciones del TAD Pila, escribir una función para realizar la copia de una pila especificada.

Ejercicio 2

Utilizando solamente las operaciones del TAD Cola, escribir una función para realizar la copia de una cola especificada.

Ejercicio 3

Utilizando solamente las operaciones del TAD Pila, implementar las funciones del TAD Cola sobre una pila especificada. Es decir, implementar las funciones **poner_en_cola** y **quitar_de_cola** que simulen el comportamiento de una cola utilizando una pila.

No se puede asumir ninguna información sobre la implementación interna de la pila, únicamente que dispone de las siguientes operaciones:

- **stack *newStack()**: crea y retorna una nueva pila vacía.

- `void push(stack *s, int x)`: inserta el elemento `x` en el tope de la pila `s`.
- `int pop(stack *s, int *x)`: elimina el elemento en el tope de la pila `s` y guarda su valor en `x`.
- `int top(stack *s, int *x)`: guarda el valor del elemento en el tope de la pila `s` en `x` sin eliminarlo.
- `int vacia(stack *s)`: retorna `1` si la pila `s` está vacía y `0` en caso contrario.

Utilizando únicamente estas operaciones, implementar las siguientes funciones:

- `void poner_en_cola(stack *s, int x)`: inserta el elemento `x` al final de la cola simulada `s`.
- `int quitar_de_cola(stack *s, int *x)`: elimina el elemento al frente de la cola simulada `s` y guarda su valor en `x`.

Ejercicio 4

Utilizando solamente las operaciones del TAD Cola, implementar las funciones del TAD Pila sobre una cola especificada. Es decir, implementar las funciones `push` y `pop` que simulen el comportamiento de una pila utilizando una cola.

No se puede asumir ninguna información sobre la implementación interna de la cola, únicamente que dispone de las siguientes operaciones:

- `queue *newQueue()`: crea y retorna una nueva cola vacía.
- `void poner_en_cola(queue *q, int x)`: inserta el elemento `x` al final de la cola `q`.

- `int quitar_deCola(queue *q, int *x)`: elimina el elemento al frente de la cola `q` y guarda su valor en `x`.
- `int frente(queue *q, int *x)`: guarda el valor del elemento en el frente de la cola `q` en `x` sin eliminarlo.
- `int vacia(queue *q)`: retorna `1` si la cola `q` está vacía y `0` en caso contrario.

Utilizando únicamente estas operaciones, implementar las siguientes funciones:

- `void push(queue *q, int x)`: inserta el elemento `x` en el tope de la pila simulada `q`.
- `int pop(queue *q, int *x)`: elimina el elemento en el tope de la pila simulada `q` y guarda su valor en `x`.

Ejercicio 5

La estructura bicola o cola de doble entrada, es una extensión del TAD Cola, al que se le puede añadir a quitar elementos por cualquier extremo, los extremos de la bicola se pueden identificar con frente y fin (al igual que la cola) y sus operaciones básicas son:

- **CrearBicola**, inicializa una bicola sin elementos.
- **BicolaVacía**, retorna true si la bicola está vacía.
- **Ponerfrente**, añade un elemento al frente.
- **PonerFinal**, añade un elemento al final.
- **QuitarFrente**, retira y retorna el elemento de frente de la bicola.
- **QuitarFinal**, retira y retorna el elemento del final de la bicola.
- **Frente**, retorna el elemento del extremo frente de la bicola.

- **Final**, retorna el elemento al extremo final de la bicola.

Implemente el TAD Bicola usando arreglos circulares.