

## Problemas Pilas y Colas

### Problema 1:

Escribir una función *Reemplazar* que tenga como argumentos una pila con tipo de elemento *enteros* y dos valores, *nuevo* y *viejo*, de forma que, si el segundo valor aparece en algún lugar de la pila, sea reemplazado por el primero.

#### Ejemplo:

Entrada:

Pila: {1, 2, 3, 2, 6}

nuevo: 45

viejo: 2

Salida: Pila: {1, 45, 3, 45, 6}

### Problema 2:

Supongamos que *TEST* es función Booleana que toma cualquier entero dado y devuelve un valor igual o distinto a cero. Consideremos además el siguiente segmento de código:

```
N=3;
pila = Pila(MAX_INT);
for (i = 1; i <= N; i++){
    if (TEST(i))
        print(i);
    else
        pila.push(i);
while (!pila.isEmpty){
    el = pila.pop();
    print(el);
}
```

¿Cuáles de las siguientes son posibles salidas del código anterior?.

- a) 1 2 3
- b) 1 3 2
- c) 2 1 3
- d) 3 1 2
- e) 2 3 1
- f) 3 2 1

### Problema 3: Paréntesis balanceados en expresión aritmética

Implementar un programa que determine si los delimitadores (, ), {, }, [, ] en una expresión aritmética (e.j. [(5+x)-(y+z)]) están equilibrados. En caso que la expresión aritmética se encuentre balanceada retornar *true*, en caso contrario retornar *false*.

Entrada: {[()]} -> Salida: true

Entrada: {[[]]} -> Salida: false

### Ejercicio 4

Se realiza sobre una pila, una secuencia de operaciones de apilar y desapilar. Las operaciones de apilar apilarán, de a uno, los números del 0 al 9 en orden (el primer apilar apilará el 0, el segundo apilar el 1, y así hasta que el décimo apilar apile el 9). Cada operación de desapilar, además de desapilar el elemento, lo imprimirá por pantalla. Determinar cuáles de las siguientes opciones no es posible que ocurra. Para cada una de las que sí son posibles, dar una secuencia de llamadas a apilar y desapilar que la cumpla.

a. 4 3 2 1 0 9 8 7 6 5

b. 4 6 8 7 5 3 2 9 0 1

c. 2 5 6 7 4 8 9 3 1 0

d. 4 3 2 1 0 5 6 7 8 9

### **Ejercicio 6**

Escribir un algoritmo que recibe una cola e invierte el orden de sus elementos. Se debe modificar la Cola recibida, no devolver una nueva ni tampoco debe usar estructuras auxiliares.

### **Ejercicio 7**

Sabiendo que se cuenta con una cola a la que se aplican las operaciones encolar, desencolar, implementar un algoritmo que permita manejar una cola con prioridad. Al desencolar debe priorizar aquellos elementos que fueron encolados con prioridad; es decir, no deben salir elementos comunes de la estructura si no salieron previamente todos los elementos con prioridad.

### **Ejercicio 8**

Implementar un algoritmo para un solitario

Se inicia con un mazo de n naipes mezclados.

Se debe apilar estos naipes en otro mazo, una arriba de otra. Sólo permite apilar un naipe si es de un número inferior al naipe que está en el tope.

Se toma el naipe que está en el tope del primer mazo, si el naipe no cumple con esa condición vuelve al final del primer mazo.

Este paso puede realizarse n veces, al finalizar, muestra la cantidad de naipes que se apilaron en el segundo mazo.

### **Ejercicio 9**

Un profesor dicta n materias y en la primera fecha de final se rinden todas en la misma aula.

Para organizar a los alumnos que rinden, nos pide un algoritmo que, dada una cola de Alumnos, devuelva para cada materia una lista con los Alumnos que rinden esa materia, respetando el orden de llegada en la cola de entrada.

### **Ejercicio 10**

Escribir un algoritmo que modele el trabajo de una torre de control de un aeropuerto con una pista de aterrizaje. Los aviones que están esperando para aterrizar tienen prioridad sobre los que están esperando para despegar. La clase debe funcionar conforme al siguiente ejemplo:

Vuelos esperando para aterrizar: AR156, AR32

Vuelos esperando para despegar: KLM1267

El vuelo AR156 aterrizó con éxito.

El vuelo AR32 aterrizó con éxito.

El vuelo KLM1267 despegó con éxito.

No hay vuelos en espera.

### **Ejercicio 11**

Escribir un algoritmo que permita modelar el funcionamiento de una impresora.

Una impresora:

- Tiene una capacidad máxima de tinta.
- Permite encolar documentos para imprimir (recibiendo el nombre de los documentos).
- Permite imprimir los documentos en el orden en que se enviaron a imprimir
- Si no hay documentos encolados, se muestra un mensaje informándolo.

- Si no hay tinta suficiente, se muestra un mensaje informándolo.
- En caso contrario, se muestra el nombre del documento, y se gasta 1 unidad de tinta.
- Permite cargar el cartucho de tinta

### **Ejercicio 12**

En un banco ingresa una cantidad  $n$  de clientes.

Cada cliente informa su DNI a la recepcionista al ingresar.

Para acelerar la atención se divide a los clientes en 2 grupos ordenados por orden de llegada:

Grupo 1: clientes con documento terminado en número par

Grupo 2: clientes con documento terminado en número impar

Se debe mostrar en una pantalla cada grupo con los DNI ordenados por orden de llegada