

# UTN – FR Mar del Plata - Técnico Universitario en Programación

## Programación I y Laboratorio I

### Trabajo Práctico N° 1: Pilas

Codificar los siguientes programas. En el caso de ser necesario se pueden utilizar pilas auxiliares.

*Inicializar en vacío es inicializar las variables internas de la pila que indican que está vacía y no tiene datos. Si no inicializo, no puedo trabajar con la pila.*

**Importante: no utilizar para la resolución de los ejercicios, variables que funcionen como contadores o acumuladores. Debe resolver la problemática aplicando pensamiento lógico.**

1. Cargar desde el teclado una pila DADA con 5 elementos. Pasar los tres primeros elementos que se encuentren en el tope a la pila AUX1 y los restantes a la pila AUX2, ambas pilas inicializadas.
2. Cargar desde el teclado la pila ORIGEN e inicializar en vacío la pila DESTINO. Pasar todos los elementos de la pila ORIGEN a la pila DESTINO.
3. Cargar desde teclado una pila DADA y pasar a la pila DISTINTOS todos aquellos elementos distintos al valor 8.
4. Cargar desde el teclado la pila ORIGEN e inicializar en vacío la pila DESTINO. Pasar los elementos de la pila ORIGEN a la pila DESTINO, pero dejándolos en el mismo orden.
5. Cargar desde el teclado la pila DADA. Invertir la pila de manera que DADA contenga los elementos cargados originalmente en ella, pero en orden inverso.
6. Pasar el primer elemento (tope) de la pila DADA a su última posición (base), dejando los restantes elementos en el mismo orden.
7. Pasar el último elemento (base) de la pila DADA a su primera posición (tope), dejando los restantes elementos en el mismo orden.
8. Repartir los elementos de la pila MAZO en las pilas JUGADOR1 y JUGADOR2 en forma alternativa.
9. Comparar la cantidad de elementos de las pilas A y B. Mostrar por pantalla el resultado.
10. Cargar las pilas A y B, y luego compararlas, evaluando si son completamente iguales (en cantidad de elementos, valores que contienen y posición de los mismos). Mostrar por pantalla el resultado.
11. Suponiendo la existencia de una pila MODELO que no esté vacía, eliminar de la pila DADA todos los elementos que sean iguales al **tope** de la pila MODELO.
12. Suponiendo la existencia de una pila MODELO (**vacía o no**), eliminar de la pila DADA todos los elementos que existan en MODELO.

# UTN – FR Mar del Plata - Técnico Universitario en Programación

## Programación I y Laboratorio I

13. Suponiendo la existencia de una pila LÍMITE, pasar los elementos de la pila DADA que sean mayores o iguales que el tope de LIMITE a la pila MAYORES, y los elementos que sean menores a la pila MENORES.

14. Determinar si la cantidad de elementos de la pila DADA es par. Si es par, pasar el elemento del tope de la pila AUX a la pila PAR y si es impar pasar el tope a la pila IMPAR. (NO contar los elementos)

15. ¿Cuál es la condición del siguiente ciclo? ¿Cuándo finaliza el ciclo? (Pila1, Pila2, y Descarte son pilas):

```
while (!pilavacia(&Pila1)) {  
    apilar (&Pila2, desapilar(&Descarte))  
}
```

16. ¿Que realiza el siguiente código escrito en lenguaje C (*Pila1*, *Aux* y *Result* son pilas):

```
while (!pilavacia(&Pila1)) {  
    if (tope(&Pila1) == 5) {  
        apilar (&Aux, desapilar(&Pila1));  
        apilar (&Result, desapilar(&Aux));  
    }  
}
```

17. Para el ejercicio “Cargar por teclado una pila ORIGEN y pasar a la pila DISTINTO todos aquellos elementos que preceden al valor 5 (elementos entre el tope y el valor 5). No se asegura que exista algún valor 5”, se realizó el siguiente programa:

// este programa carga por teclado una pila Origen y pasa a la pila Destino todos aquellos elementos que preceden el valor 5

```
int main() {  
    Pila Origen, Distinto;  
    inicpila(&Origen);  
    inicpila(&Distinto);  
    leer(&Origen);  
    if (!pilaVacia(&Origen)){  
        while (tope(&Origen) != 5) {  
            apilar (&Distinto, desapilar(&Origen));  
        }  
    }  
}
```

- ¿Resuelve el problema planteado?
- ¿Cuáles son los errores que encuentra?

# UTN – FR Mar del Plata - Técnico Universitario en Programación

## Programación I y Laboratorio I

- c. Reescribir el código para que resuelva adecuadamente el problema planteado.
- d. Indicar los componentes del programa.

18. Dado el siguiente ciclo (*Pila1*, *Pila2* y *Descarte* son pilas):

```
while ( (!pilaVacia(&Pila1)) && (!pilaVacia(&Pila2)) ) {  
    apilar (&Descarte, desapilar(&Pila1));  
    apilar (&Descarte, desapilar(&Pila2));  
}
```

- a. ¿Cuál es la condición del ciclo? Explique con sus palabras
- b. ¿Cuales son los posibles estados de ambas pilas al finalizar el ciclo?