

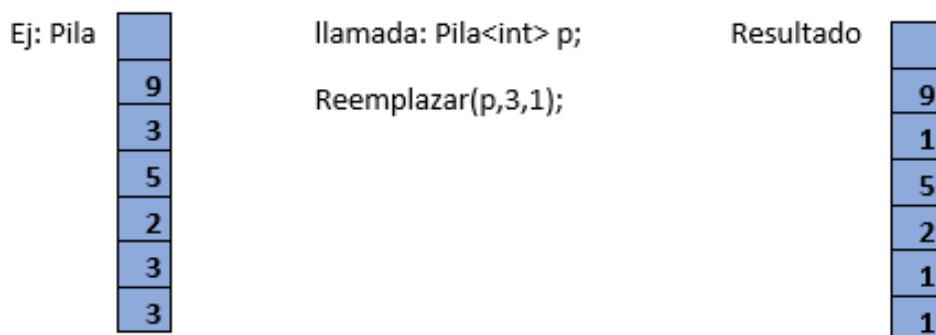
Unidad 3 – Pilas

Ejercicio N°1

Escriba un programa que introduzca una palabra y utilice una pila para imprimir la misma palabra invertida.

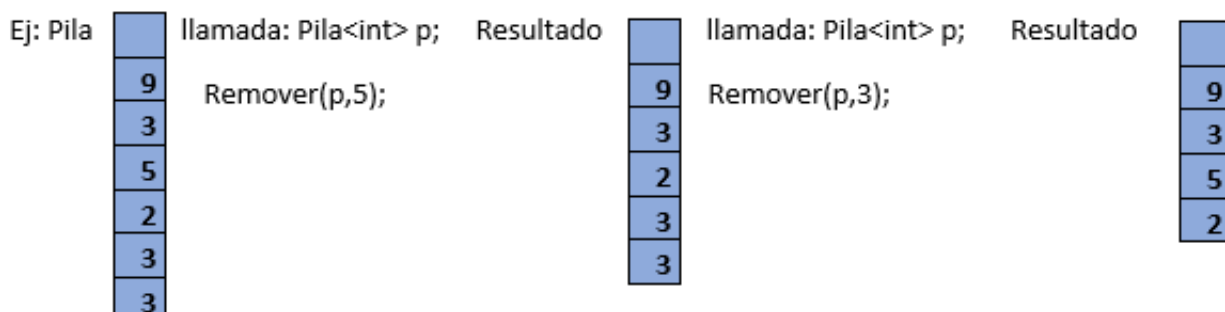
Ejercicio N°2

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



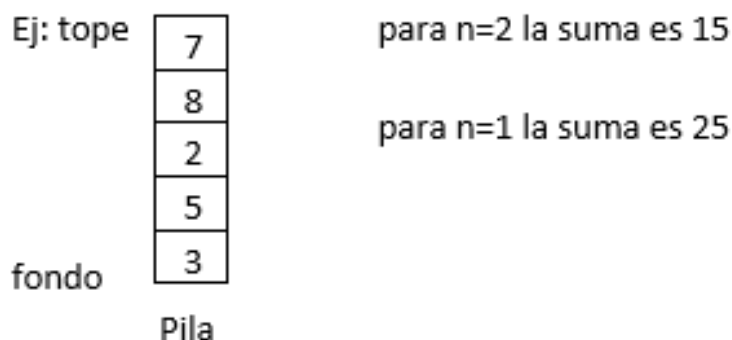
Ejercicio N°3

Crear una función que reciba una pila por referencia y un número. La función debe sacar la primera ocurrencia de dicho valor de la pila sino se repite el valor en la pila, de lo contrario debe eliminar todos los valores repetidos a excepción de la primera ocurrencia. Si el valor no se encuentra en la pila muestra un mensaje indicando que no pertenece a la pila. Imprimir la pila al final para verificar si hubo o no modificaciones.



Ejercicio N°4

Crear un programa que solicite números enteros al usuario hasta que desee, almacenar los números en una pila y pedir también un número entero "n". Luego crear una función que reciba una pila y un número entero "n", si el "n" se encuentra en una de las posiciones de la pila, devuelve la suma de los números en cada una de las posiciones de la pila hasta llegar a "n" (sin incluir a n), si el número "n" no se encuentra en la pila, la función devuelve la suma de todos los elementos de la pila. Imprimir primero la pila antes de sumar sus elementos.



Ejercicio N°5

Crear un programa e implementar una función que reciba una pila de números enteros y dado un número "n" por el usuario devuelva una nueva pila con la suma de cada uno de los elementos de la pila original más el n dado por el usuario. Los elementos en la nueva pila deben tener el orden que tenían en la pila original, y la pila original debe quedar en el mismo estado a la inicial, mostrar por pantalla ambas pilas.

Por ejemplo, se tiene la pila de enteros [1, 2, 3, 6, 2] (tope es el número 2), y la función sumarNro recibe como parámetros la pila y n=1 (devuelve la suma entre el número 1 y el número recibido en cada posición de la pila original), la pila resultante debe ser [2, 3, 4, 7, 3] (el tope es el número 3)

Ejercicio N°6 : Control de Desplazamientos en un Edificio

Contexto: Un edificio cuenta con un sistema de control de accesos a través de tarjetas magnéticas. Cada vez que una persona entra o sale del edificio, su tarjeta se registra en el sistema. Queremos implementar un programa que, utilizando una pila, simule el movimiento de una persona dentro del edificio.

Enunciado:

Implementa un programa en C++ que utilice una pila para simular el movimiento de una persona en un edificio. Cada vez que la persona entra en un área del edificio (por ejemplo, "Recepción", "Sala de Reuniones", "Oficina"), se debe apilar el nombre del área. Cuando la persona sale del área, se debe desapilar el nombre. Al final del día, el programa debe mostrar el recorrido de la persona en orden inverso al que lo hizo (último lugar visitado al primero).

Entrada:

Serie de operaciones de entrada ("in") o salida ("out") de áreas específicas del edificio.

Salida:

El recorrido completo de la persona en orden inverso, mostrando los lugares que visitó.

Ejemplo: Si la persona visitó las áreas en el siguiente orden:

Entra a "Recepción"

Entra a "Oficina"

Entra a "Sala de Reuniones"

Sale de "Sala de Reuniones"

Entra a "GYM"

El programa debería mostrar:

"GYM"

"Oficina"

"Recepción"

Ejercicio N°7: Deshacer Operaciones en un Editor de Texto

Contexto: Imagina que estás utilizando un editor de texto que tiene una función de "deshacer" para revertir los cambios recientes. Cada vez que realizas un cambio en el documento, el editor lo guarda para que puedas deshacerlo si es necesario. Implementa esta funcionalidad usando una pila.

Enunciado:

Escribe un programa en C++ que simule la funcionalidad de "deshacer" de un editor de texto utilizando una pila. Cada vez que el usuario realiza una acción (por ejemplo, escribir una palabra, borrar una palabra), se debe apilar esta acción. Cuando el usuario decide "deshacer" una acción, se debe desapilar y mostrar cuál fue la última acción realizada.

Entrada:

Serie de acciones como "escribir" o "borrar" seguidas de la palabra correspondiente.

Comando "deshacer" para revertir la última acción.

Salida:

Estado actual del texto después de cada operación.

Acción revertida en cada comando de "deshacer".

Ejemplo: Si el usuario realiza las siguientes acciones:

Escribir "Hola"

Escribir "Mundo"

Borrar "Mundo"

Deshacer

Deshacer

El programa debería mostrar:

Texto actual: "Hola"