



Universidad Tecnológica
Centroamericana

UNITEC

Laboratorio de Programación III

Victor Euceda

Examen 2



Objetivos

- Utilizar los flujos estándares de impresión y obtención de datos.
- Aplicar diferentes funciones estándares y estructuras de control para la resolución de problemas matemáticos
- Modelar soluciones a problemas mediante funciones y TDA en C++

Desarrollo de la práctica

Para el desarrollo satisfactorio de los ejercicios, siga las instrucciones que a continuación se le presentan, después de cada ejercicio encontrará preguntas que deberá desarrollar en base al ejercicios elaborado.

Ejercicio 1

Deberá implementar una pila ,utilizando apuntadores, para el cálculo de operaciones aritméticas basándose correctamente en la precedencia de operadores.

Para el desarrollo de la práctica se ingresará una operación que debe leerse como cadena y luego ser ingresada a la pila correctamente, a medida que se realicen las operaciones se debe mostrar el valor que se va sacando de la pila y la operación que se está realizando.

Las operaciones que se deben implementar son:

- ☐ Suma y Resta: $1+2-4+3-5$
- ☐ Multiplicación: $1*2*-4$
- ☐ División: $3/4$
- ☐ Agrupación: $(1+2)/4$

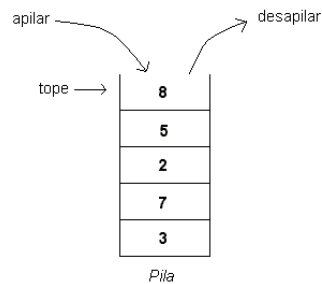
□ Asignación: $A = 1+2*4$

□ Para la asignación se debe guardar el identificador y el valor numérico para poder utilizarlo en siguientes operaciones.

Deberá implementar la lectura de archivos de texto, que sea capaz de realizar lecturas de operaciones y cargarlas correctamente a la pila.

TDA pila

Una *pila* (*stack* o *pushdown* en inglés) es una lista de elementos de la cual sólo se puede extraer el último elemento insertado. La posición en donde se encuentra dicho elemento se denomina *tope* de la pila. También se conoce a las pilas como *listas LIFO* (LAST IN - FIRST OUT: el último que entra es el primero que sale).



La interfaz de este TDA provee las siguientes operaciones:

- **apilar(x)**: inserta el elemento x en el tope de la pila (**push** en inglés).
- **desapilar()**: retorna el elemento que se encuentre en el tope de la pila y lo elimina de ésta (**pop** en inglés).
- **tope()**: retorna el elemento que se encuentre en el tope de la pila, pero sin eliminarlo de ésta (**top** en inglés).
- **estaVacia()**: retorna *verdadero* si la pila no contiene elementos, *falso* en caso contrario (**isEmpty** en inglés).

Ponderación

Elemento	Puntaje
Implementación de Pila	7
Implementación de Operaciones	5
Implementar Asignación	2
Makefile e Interfaz	1

Especificaciones de entrega

Deberá subir a edmodo un archivo de texto con su nombre, número de cuenta, usuario de GitHub y la dirección del repositorio de GitHub.