**PILAS**

**Fundamentos teóricos:**

Una pila es una estructura de datos de entradas ordenadas tales que solo se introduce y elimina por un extremo, llamado cima o tope.

Una pila (stack) es una colección ordenada de elementos a los que solo se puede acceder por un único lugar o extremo de la pila.

Los elementos de la pila se añaden o se quitan (borran) de la misma solo por su parte superior (cima). Este es el caso de una pila de platos, una pila de libros, etc.

Cuando la pila se dice que esta ordenada, lo que se quiere decir es que hay un elemento al que se puede acceder primero (el que está encima de la pila), otro elemento al que se puede acceder en segundo lugar (justo el elemento que esta debajo de la cima), un tercer, etc.

No se requiere que las entradas se puedan comparar utilizando el operador <<menor que (<) >> y pueden ser de cualquier tipo.

Las entradas de la pila deben ser eliminadas en el orden inverso al que se situaron en la misma. Se puede crear una pila de libros, situando primero un diccionario, encima una enciclopedia y encima de ambos una novela en la parte superior.

Debido a su propiedad especifica <<último en entrar-primero en salir>> se conoce a las pilas como estructura de datos LIFO (last in, first out).

Las operaciones usuales en la pila son insertar y quitar. La operación insertar (push) añade un elemento en la cima de la pila y la operación quitar (pop) elimina o saca un elemento de la pila.

La pila se puede implementar mediante arrays en cuyo caso su dimensión o longitud es fija, y mediante punteros o listas en cuyo caso se utiliza memoria dinámica y no existe limitación en su tamaño.

Una pila puede estar vacía (no tiene elemento) o llena (en el caso de tener tamaño fijo, si no caben más elementos en la pila).

Si un programa intenta sacar un elemento de una pila vacía, se producirá un error debido a que esa operación es imposible, esta situación se denomina desbordamiento negativo (underflow).

Por el contrario, si un programa intenta poner un elemento en una pila, se produce un error llamado desbordamiento (overflow) o rebosamiento. Para evitar estas situaciones se diseñan funciones, que comprueban si la pila está llena o vacía.

**Definición**

Las pilas son estructuras de datos lineales, como los arreglos, ya que los componentes ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y predecesor, con excepción del ultimo y del primero, respectivamente.

Una pila se define formalmente como una colección de datos a los cuales de puede acceder mediándote un extremo, que se conoce generalmente como tope.

Las pilas no son estructuras fundamentales de datos. Para su representación requiere el uso de otras estructuras de datos, como arreglos o listas.

**Operaciones básicas:**

* Insertar un elemento
* Obtener elemento (tope)
* Eliminar elemento
* Está vacía (comprobación)
* Recorrer pila (tope al final)

Para analizar las operaciones de una pila se deben crear tres elementos que permitan implementar la pila:

* **Clase nodo:** esta clase representa un nodo individual en la estructurade datos y contiene información sobre el valor almacenado en el nodo y punteros a los nodos vecinos.
* **Clase pila:** esta clase representará las operaciones que se pueden realizar con las pilas.
* **Método main:** método principal en donde se creará una instancia de la pila para acceder a sus operaciones.

**APLICACIONES DE LAS PILAS**

**Llamadas a subprogramas**

Cuando se tiene un programa que llama a un subprograma, modulo o función, internamente se usan pilas para guardar el estado de las variables del programa y las instrucciones pendientes de ejecución.

Cuando se termina la ejecución del subprograma, los valores almacenados en la pila se recuperan para continuar con la ejecución del programa en el punto en el cual fue interrumpido.

Además de las variables se recupera la dirección del programa en la que hizo la llamada, porque a esa posición se regresa el control del proceso.

**Recursividad**

Utiliza una pila, donde se van almacenando las instrucciones a ejecutar con los valores de las constantes y las variables en ese momento.

Cuando se termina la ejecución se llega al estado básico, se toma la instrucción del tope de la pila y se continúa ejecutando, hasta que la pila quede vacía.

**Balanceo o equilibrio de símbolos**

También pueden utilizarse las pilas para revisar programas en busca de errores como por ejemplo en el caso de los símbolos.

En cada compilador se utilizan las pilas para comprobar si se han producido errores, para que cada paréntesis, llave o corchete de apertura tenga su correspondiente cierre. Es decir, que un programa presenta más símbolos de apertura que de clausura, el programa tendría errores y no sería valido.

También debemos tener en cuenta el orden en el que estén dichos símbolos, porque puede que, aunque tenga los mismos símbolos no estén colocados correctamente.

Utilizar una pila para introducir símbolos abiertos. Cuando se recibe un símbolo cerrado, se compara con el que está en la cima de la pila (si es que hay alguno), si son de distinto tipo es que no es correcto, y si no son del mismo tipo se elimina y se sigue leyendo.

**Tratamiento de expresiones aritméticas**

Nosotros generalmente usamos expresiones como la siguiente: **a+b,** este tipo de notación es llamado notación **fija**, puesto que el operador se encuentra entre los dos operandos. Dependiendo en la posición del operador se pueden tener otros dos tipos de notación:

**Prefija:** el operador se ubica antes de los operadores **+ab**

**Postfija:** el operador se ubica después de los operadores **ab+**

**Infija:**

Una expresión postfija se puede evaluar con la ayuda de una pila usando el siguiente procedimiento:

Recorrer la expresión de izquierda a derecha y para cada término hacer:

* Si el termino es un valor, introducirlo en una pila

Si el termino es un operador:

* Sacar dos operadores de la pila
* Aplicar el operador
* Meter el resultado en la pila