

2.3 Pilas, explicación y ejemplos.

Una Pila es una estructura de datos, que simula el acto humano de encimar cosas para tenerlas controladas o para algún servicio. Por ejemplo, se tienen cuatro cajas de mercancías colocadas una sobre otra formando una pila, como se muestra:

Caja 4 ← la parte más alta, la cima, o el tope de la pila

Caja 3

Caja 2

Caja 1

==== piso

La parte más alta de la pila se denomina la cima o el tope de la pila.

En memoria RAM la idea es simular una pila formada con datos en la memoria, se puede usar un arreglo como base.

Se usará un arreglo de cierto tipo como base.

Tipo base [tamaño]; entero tamaño = 20;

// Tipo puede ser: entero (int), real (doublé), carácter (char), una cadena de caracteres, o una clase definida por el usuario.

Se usará un apuntador denominado tope, a la parte más alta de la pila, por analogía con una pila de cosas el arreglo se pondrá vertical.

Al inicio la pila está vacía, tope=0, si se agrega un elemento a la pila el tope aumenta en 1, se agregan elementos a la pila de uno en uno, si se retira un elemento de la pila, el tope decrece en uno.

Base [5]

Base [4] = T

Base [3] = X

Base [2] = G

Base [1] = Ñ tope=4

Las declaraciones para una pila son:

Entero tamaño=30; entero base [tamaño]; entero tope = 0; indicando que la pila está vacía

El algoritmo para agregar cosas a una pila se llama push (dato), su código es:

Push (dato)

```
si tope = tamaño
    Escribe "PILA LLENA"
else tope=tope+1
    base [tope]=dato
```

fin

El algoritmo para sacar cosas de la pila se llama pop () y se tiene

```
tipo pop ()
si tope=0
    Escribe "PILA VACIA"
else valor=base[tope]
    tope = tope -1
    regresa valor
```

fin.

Una rutina de servicio que permite obtener una instantánea de la pila, denominada estado actual es:

```
estado actual ()
if (tope=0)
    Escribe "PILA VACÍA"
else {k=tope;
    while (k>0)
        {escribe base[k];
        k=k-1;}}
}
```