

Pilas

Una pila es una estructura de datos en la que todas las inserciones y eliminaciones se realizan por un extremo denominado cima de la pila. Una analogía es la pila de platos o una pila de cajas.

La implementación de una pila se puede realizar mediante arrays o con punteros. El inconveniente de la implementación de una pila con un array es que su tamaño máximo se debe especificar en tiempo de compilación.

En una pila el último elemento añadido es el primero en salir de la pila. Es por esa razón que a las pilas se las denomina también listas Lifo (Last input first output. “último en entrar, primero en salir”).

Las operaciones básicas que definen la Pila son:

- Pilavacia(P) Crea una pila sin elementos.
- Esvacia(P) Devuelve verdadero si la pila P no tiene elementos.
- Cima(P) Devuelve el elemento que está en la cima de la pila.
- Suprime(P) Elimina el elemento que está en la cima de la pila.
- Sacar(X, P) Devuelve el elemento cabeza y lo suprime.
- Meter(X, P) Añade el elemento X en la pila.

Implementación del TAD pila con arrays

Un array constituye el depósito de los elementos de la pila. el rango del array debe ser lo suficientemente amplio para poder contener el máximo previsto de elementos de la pila. Un extremo del array se considera el fondo de la pila, que permanecerá fijo. La parte superior de la pila, cima, estará cambiando dinámicamente durante la ejecución del programa. Además del array, una variable entera nos sirve para tener en todo momento el índice del array que contiene el elemento cima. Las declaraciones, procedimiento y funciones para representar el TAD pila forman parte de la unidad pilas.

Utilización de una pila.

“Se desea invertir una cadena de caracteres y se trata de determinar si han sido invertidos correctamente dichos caracteres. Para ello se lee la cadena de caracteres; a continuación, en la siguiente línea los caracteres”

La estrategia a seguir es sencilla, los caracteres de la cadena se almacenan en una pila. Según se van leyendo los caracteres de la cadena invertida se comparan con el carácter cima de la pila; de este modo se aprovecha la principal característica de una pila: último en entrar primero en salir. El último carácter introducido debe ser igual que el primer carácter de la cadena invertida.

```

function Esinversa: boolean;
var
    Ch: char;
    Pila: Tipopola;
    Inv: boolean;
begin
    Pilavacia(Pila);
    writeln("caracteres de la cadena: ");
    repeat
        read(Ch);
        Meter(Ch, Pila)
    until eoln;
    readln;
    {los caracteres leídos son comparados con la pila}
    Inv:= true;
    while not Esvacia(Pila) and Inv do
        begin
            read(Ch);
            Inv:= Ch= Cima(Pila);
            Suprimir(Pila);
        end;
        {consideramos que si hay más carácter que en la pila, no ha sido
        invertida correctamente}
    Esinvera:= Inv and eoln;
    readln
end;

```

Para sacar los elementos de la pila se puede seguir la alternativa de utilizar otra variable carácter D y la operación Sacar:

```

while not Esvacia(Pila) and Inv do
    begin
        read(Ch);
        Sacar(D, Pila);
        Inv:= Ch = D;
    end;

```

Referencia:

Luis Joyane Aguilar, Ignacio Zahonero Martínez. (1998). Estructura de datos. España: McGraw-Hill.