

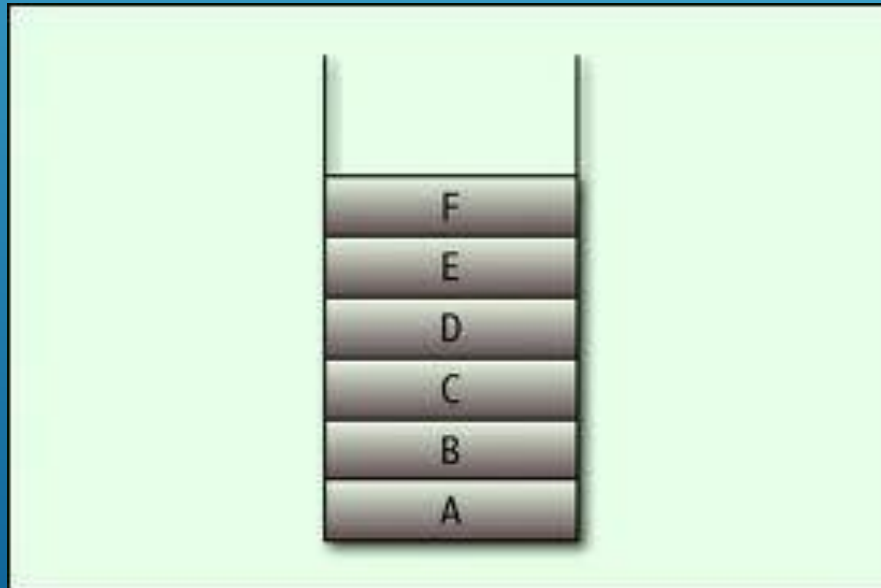
Pilas

Que es una pila ??



Pilas

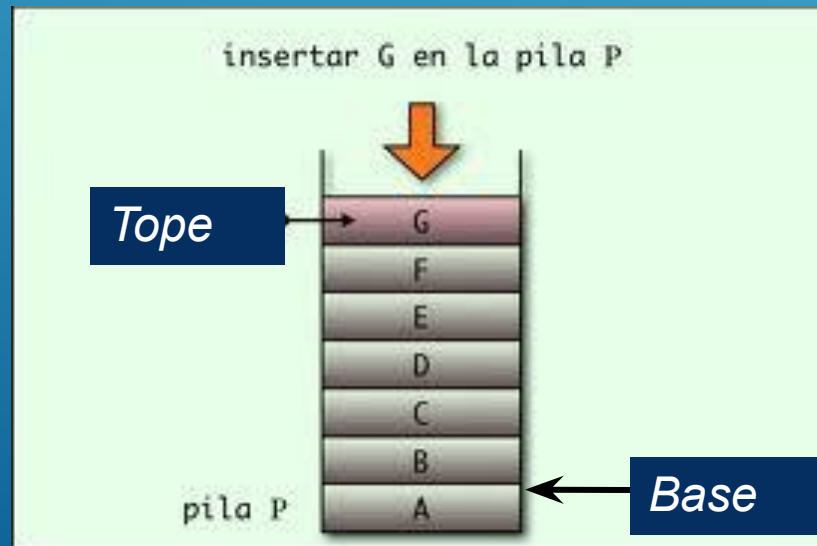
Una pila es una lista de elementos en la que se pueden insertar y eliminar elementos sólo por uno de los extremos.



Pilas

La inserción como la extracción de elementos de una Pila se realizan en el tope. Este comportamiento lleva a que el modelo también se llame

LIFO ("last in, first out" = "último en entrar primero en salir").



Pilas

Como consecuencia, los elementos de una pila serán eliminados en orden inverso al que se insertaron. Es decir, el último elemento que se metió a la pila será el primero en salir de ella.



Pilas

Que podemos hacer con una pila ??

Podemos :

- ☐ *Crearla*
- ☐ *Inicializarla*
- ☐ *Agregar un elemento*
- ☐ *Sacar un elemento*
- ☐ *Ver si esta vacía*
- ☐ *Ver el elemento que esta en el tope*
- ☐ *Mostrar el contenido de toda la pila*



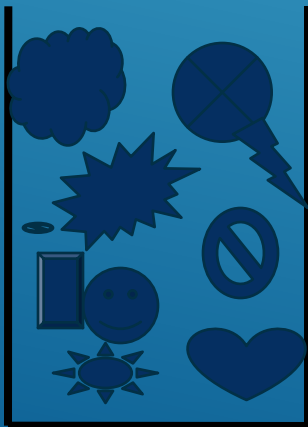
Pilas

¿ Como creamos una pila ?

☐ ***Pila idPila;***

Crea una pila.

Ej.: Crea una pila de nombre “pilita”.



pilita

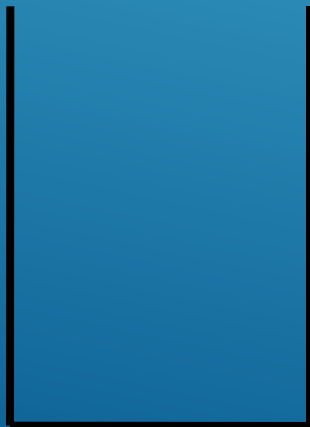
Pila pilita;



Pilas

¿ Como inicializamos una pila ?

- ☐ *inicpila (&idPila);*
- ☐ *Inicializa a la pila con cero elementos.*



pilita

- ☐ *inicpila (&pilita);*



Pilas

❑ ¿ Como agregamos un elemento a una pila ?

Tenemos dos maneras:

- ❑ *apilar (&idPila, número);*
 - ❑ *Pone el elemento <número> al tope de la pila.*
- ❑ *leer (&idPila);*
 - ❑ *Agrega al tope de la pila un elemento ingresado desde el teclado por el usuario.*

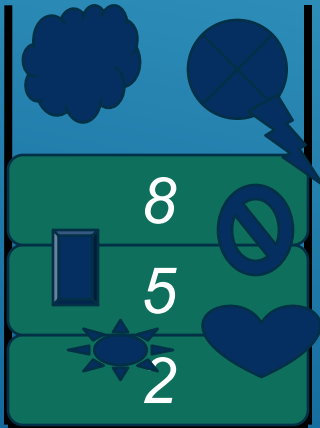


Pilas

Ejemplo 1.1:

Crear una pila y cargarla con 3 elementos.

pilita



Pila pilita;

inicpila(&pilita);

apilar(&pilita, 2);

apilar(&pilita, 5);

apilar(&pilita; 8);



Pilas

Ejemplo 1.1

// incluir librerías

```
int main(){  
    Pila pilita;  
    inicpila (&pilita);  
    apilar(&pilita, 2);  
    apilar(&pilita, 5);  
    apilar(&pilita, 8);  
}
```



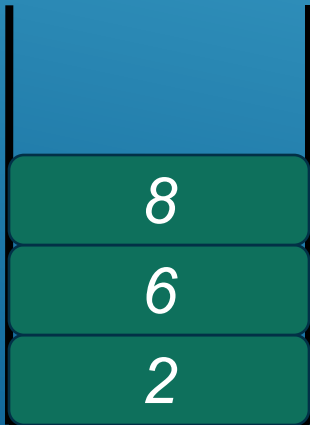
Pilas

Ejemplo 1.2:

Crear una pila y cargarla con 3 elementos desde el teclado



Pila



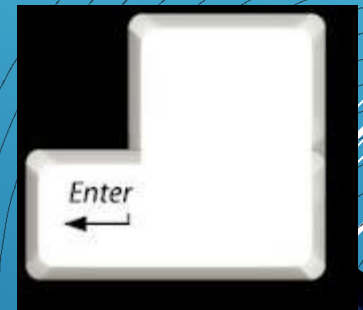
Pila pilita;

inicpila(&pilita);

leer(&pilita);

leer(&pilita);

leer(&pilita);



Pilas

Ejemplo 1.2

// incluir librerías

```
int main(){  
    Pila pilita;  
    inicpila(&pilita);  
    leer(&pilita);  
    leer(&pilita);  
    leer(&pilita);  
}
```



Pilas

❑ ¿ Como sabemos que hay en el tope ?

❑ ***tope(& idPila);***

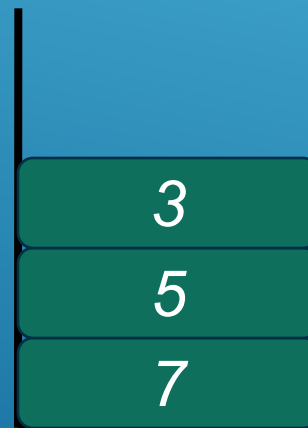
- ❑ *Devuelve el valor del elemento que está en el tope de la pila.*
- ❑ *No lo elimina de la pila.*



tope(&pilita);



pilita



tope(&pilota);



pilota



Pilas

❑ *¿ Como sacamos un elemento de una pila ?*

❑ *desapilar (&idPila); → <número>*

Devuelve el elemento que está en el tope de la pila y lo elimina de la misma.

Luego de llamar a desapilar, el elemento que estaba después del elemento extraído pasará al tope o la pila se vaciará si éste era el último.

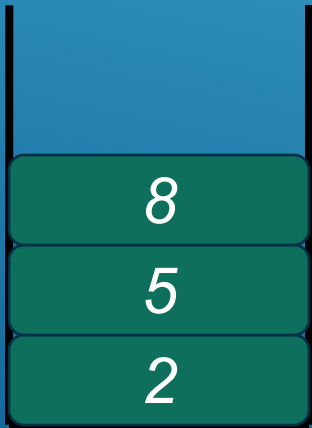
Si se llama a desapilar con una pila vacía como parámetro se producirá un error y se abortará el programa



Pilas

Ejemplo 1.3:

Desapilar dos veces una pila cargada con tres elementos.



pilita

desapilar(&pilita);

desapilar(&pilita);



Ejemplo 1.4 __ Leer tres números desde el teclado en “pilita”, apilar dos números (8 y 6) en “pilota”. Pasar el tope de “pilita” y luego el tope de “pilota” a una pila llamada “destino”.

```
int main () {  
    Pila pilita,pilota,destino;  
    inicpila(&pilita);  
    inicpila(&pilota);  
    inicpila(&destino);  
    leer(&pilita);  
    leer(&pilota);  
    leer(&pilota);  
    apilar(&pilota, 8);  
    apilar(&pilota, 6);  
    apilar(&destino, desapilar(&pilota));  
    apilar(&destino, desapilar(&pilota));  
}
```



Ejemplo 1.4: Leer tres números desde el teclado en “pilita”, apilar dos números (8 y 6) en “pilota”. Pasar el tope de “pilita” y luego el tope de “pilota” a una pila llamada “destino”.

```
int main () {
```

```
    Pila pilita, pilota, destino;
```

```
    inicpila(&pilita);
```

```
    inicpila(&pilota);
```

```
    inicpila(&destino);
```

```
    leer(&pilita);
```

```
    leer(&pilita);
```

```
    leer(&pilita);
```

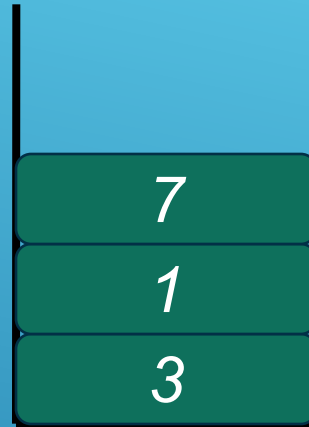
```
    apilar(&pilota, 8);
```

```
    apilar(&pilota, 6);
```

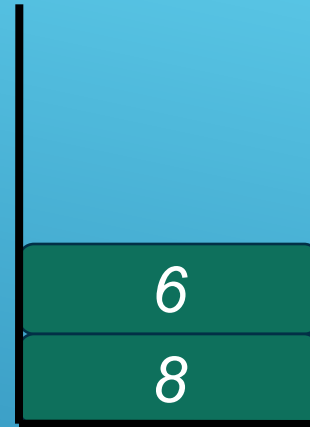
```
    apilar(&destino, desapilar(&pilita));
```

```
    apilar(&destino, desapilar(&pilota));
```

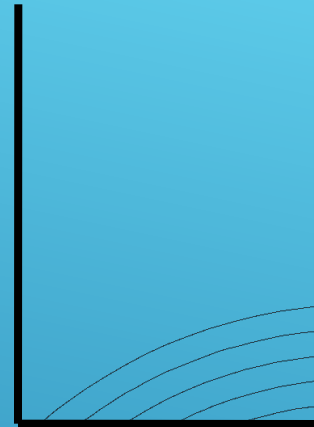
```
}
```



pilita



pilota



destino

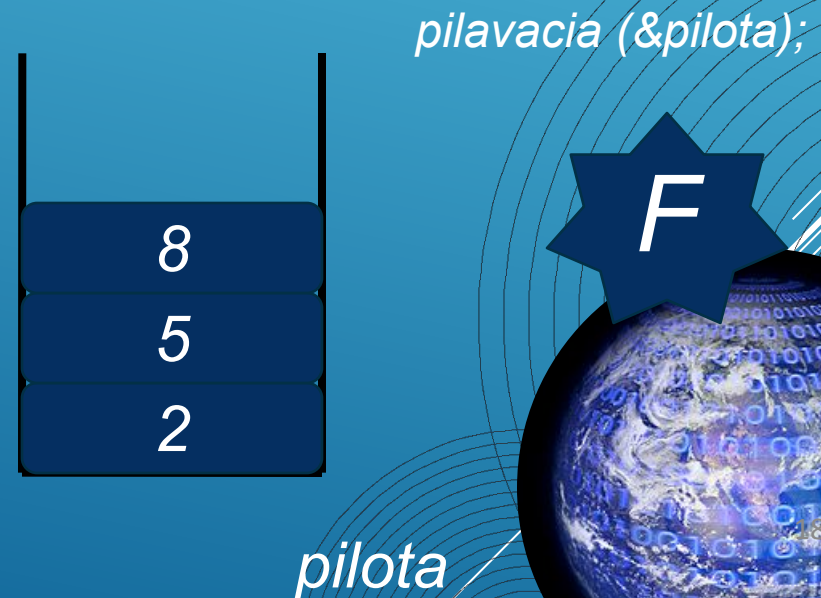
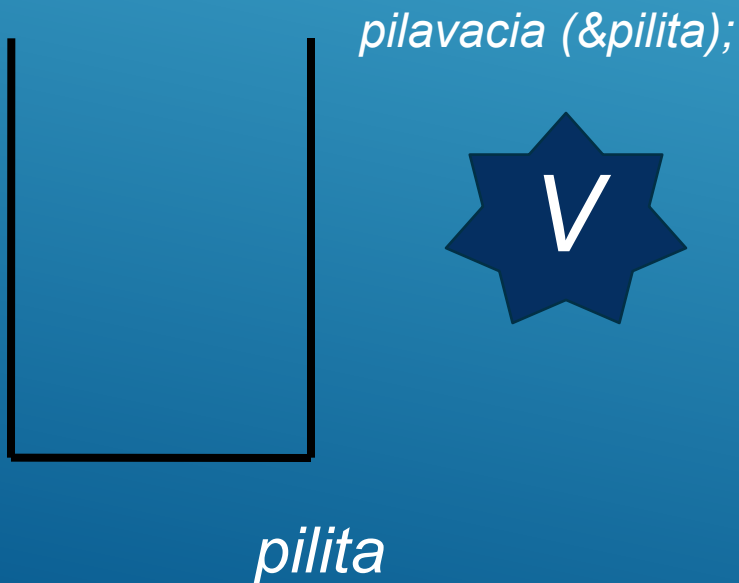


Pilas

❑ ¿ Como sabemos si una pila esta vacía ?

❑ *`pilavacia(&idPila);`*

Me dice si es Verdadero o Falso que la pila este vacia.

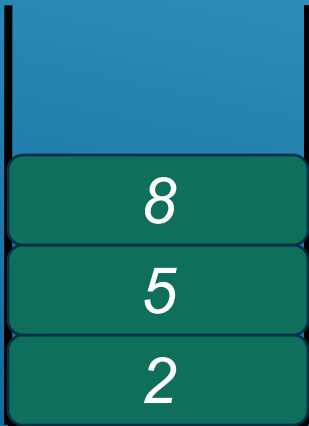


Pilas

❑ ¿ Como vemos el contenido de una pila ?

❑ *mostrar(&idPila);*
Ej: mostrar(&pilita);

Pilita



Ejemplo 1.5 __ Leer tres números desde el teclado en “pilita”, Si el tope de pilita es mayor que 5 pasarlo a “pilota”. Mostrar ambas pilas por pantalla.

```
Int main () {  
    Pila pilita, pilota;  
    inicipila(&pilita);  
    inicipila(&pilota);  
    leer(&pilita);  
    leer(&pilita);  
    leer(&pilita);  
    if ( tope(&pilita) > 5) {  
        apilar(&pilota, desapilar(&pilita));  
    }  
    mostrar(&pilita);  
    mostrar(&pilota);  
}
```



Ejemplo 1.5 __ Leer tres números desde el teclado en “pilita”, Si el tope de pilita es mayor que 5 pasarlo a “pilota”. Mostrar ambas pilas por pantalla.

```
int main () {
```

```
    Pila pilita, pilota;
```

```
    inicpila (&pilita);
```

```
    inicpila(&pilota);
```

```
    leer(&pilita);
```

```
    leer(&pilita);
```

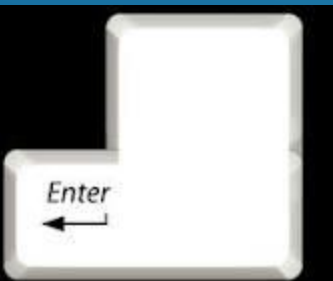
```
    leer(&pilita);
```

```
    if (tope(&pilita)>5) {  
        apilar(&pilota, desapilar(&pilita);
```

```
    }  
    mostrar(&pilita);
```

```
    mostrar(&pilota);
```

```
}
```



Ejemplo 1.6 __ Leer una pila “Origen” con 3 números. Pasar el tope de “Origen” a “Destino1” si es mayor que 10, de lo contrario pasarlo a “Destino2”.

```
int main () {  
    Pila origen, destino1, destino2;  
    inicpila(&origen);  
    inicpila(&destino1);  
    inicpila(&destino2);  
    apilar(&origen, 15);  
    apilar(&origen, 7);  
    apilar(&origen, 8);  
    apilar(&origen, 20);  
    if ( tope(&origen) > 10) {  
        apilar(&destino1, desapilar(&origen));  
    } else {  
        apilar(&destino2, desapilar(&origen));  
    }  
}
```



Ejemplo 1.6 __ Leer una pila “Origen” con 3 números. Pasar el tope de “Origen” a “Destino1” si es mayor que 10, de lo contrario pasarlo a “Destino2”.

```
int main ( ) {
```

```
    Pila origen,destino1, destino2;
```

```
    inicpila(&origen);
```

```
    inicpila(&destino1);
```

```
    inicpila(&destino2);
```

```
    leer(&origen);
```

```
    leer(&origen);
```

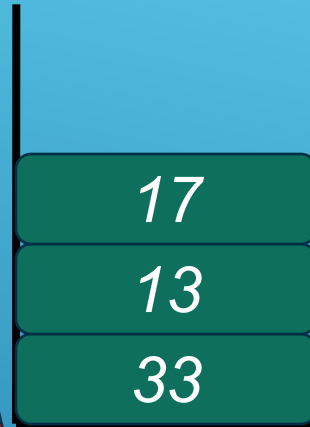
```
    leer(&origen);
```

```
    if (tope(origen)>10) {
```

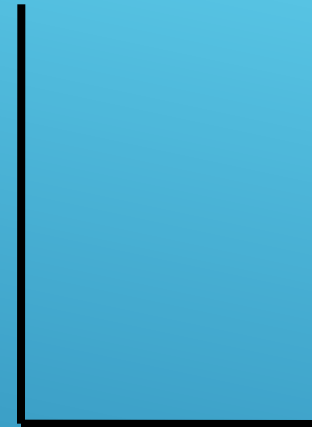
```
        apilar(&destino1, desapilar(&origen));
```

```
    } else {
```

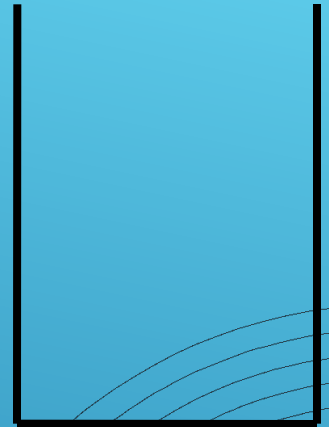
```
        apilar(&destino2, desapilar(&origen));
```



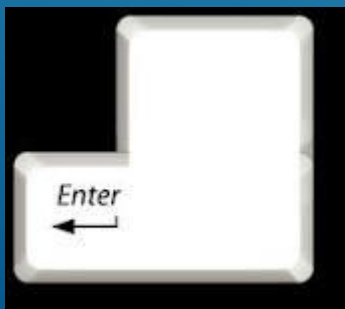
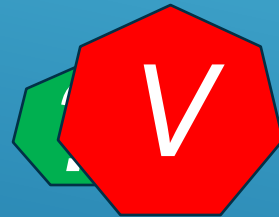
origen



destino1



destino2



Pilas... como recorrerlas?

Ejemplo 1.8 __ Leer tres números desde el teclado en “pilota”. Pasarlos todos a “pilota” utilizando una estructura de repetición.

```
int main ( ) {  
    Pila pilota, pilota;  
    inicpila(&pilota);  
    inicpila(&pilota);  
    leer(&pilota);  
    leer(&pilota);  
    leer(&pilota);  
    while ( ! pilavacia(&pilota)) {  
        apilar(&pilota, desapilar(&pilota));  
    }  
}
```



Ejemplo 1.8 __ Leer tres números desde el teclado en “pilita”. Pasarlos todos a “pilota” utilizando una estructura de repetición.

```
int main ( ) {  
    Pila pilita, pilota;  
    inicpila(&pilita);  
    inicpila(&pilota);  
    leer(&pilita);  
    leer(&pilita);  
    leer(&pilita);
```

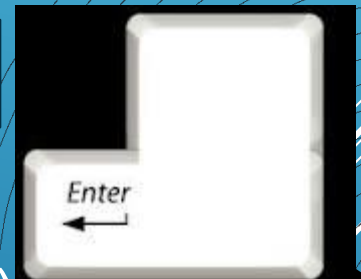
```
    while (! pilavacia)  
        apilar(&pilota, des pilavar(&pilita))  
}  
}
```

pilita

8
6
2



V ??????? **F**



F ?????? **V**

