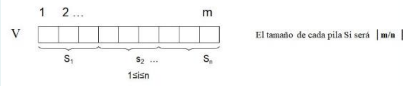


## Pregunta 1

Finalizada

Se puntúa  
80.00 sobre  
100.001<sup>er</sup> Marcar  
pregunta

El esquema para implementar múltiples Pilas en un vector  $V$  es el siguiente:



Se utilizarán dos arreglos adicionales

$B(1:n+1)$  para almacenar el inicio y el fin de las pilas

$S_i = B(i)$  a  $B(i+1)$  tamaño de cualquier  $S_i$

$T(1:n)$  para almacenar el estado del puntero Top de cada pila  $S_i$

Al crearlas  $n$  pilas el estado de  $B$  y  $T$  será:

$B(i) = T(i) = \lfloor m/n \rfloor (i-1)$   $1 \leq i \leq n$

Para  $B(n+1) = m$  por si existe división fragmentada de  $\lfloor m/n \rfloor$

Se requiere implementar  $n$  Doble Pilas en el vector  $V$  del esquema mostrado arriba, una DoblePila se caracteriza porque aunque los elementos pueden ser insertados solamente por la cima de la pila, pueden ser extraída tanto por la cima como por el fondo de la pila, las operaciones básicas son las siguientes:

Funciones

1. `Create()`  $\rightarrow$  Pila // Crea una DoblePila
2. `IsVacia(i)`  $\rightarrow$  Boolean // Es TRUE si la Pila  $i$  esta vacía, de lo contrario es FALSE. (10 pts)
3. `IsFull(i)`  $\rightarrow$  Boolean // Es TRUE si la Pila  $i$  esta llena, de lo contrario esta vacía. (10 pts)
4. `LongEle(i)`  $\rightarrow$  N // Cuantos elemento tiene la Pila  $i$  (20 pts)
5. `Add(i,x)`  $\rightarrow$  Pila // Adiciona el elemento  $x$  a la Pila  $i$  (10 pts)
6. `DelTopN(i,n)`  $\rightarrow$  Pila // Borra los  $n$  elemento tope de la Pila  $i$ , la pila debe tener mayor o igual cantidad de elementos que  $n$  (25 pts)
7. `DelFonN(i,n)`  $\rightarrow$  Pila // Borra los  $n$  elemento del fondo de la Pila  $i$ , la pila debe tener mayor o igual cantidad de elementos que  $n$  (25 pts)

Implementar las funciones citadas en una clase denominada `clsDPila.cs` utilizando un solo Vector en el lenguaje C# (pueden utilizar como referencia `clsPila.cs`)

Los nombres de las funciones y parámetros de entrada y salida deben coincidir con la especificación.

Enviar a la plataforma virtual la clase `clsDPila.cs`

```
Add(i,x)
If IsFull(i)==false
{Top[i]=++
S[Top[i]]=x}
DelTopN(i,n)
If Long(i) >= n
{ For j=1 to n
Top[i]=--}
DelFonN(i,n)
If Long(i) >= n
{For j=1 to n
Rear[i]=++}
Vacia(S)
return Top[i]=Rear[i]
Llena(i)
return Top[i]=B[i+1]
Long(i)
Return (T(i)-B(i))-(R(i)-B(i))
```

Comentario: