

## PRÁCTICA/LABORATORIO N° 02

### ARREGLOS

#### Objetivos:

- Utilizar [Dev-C++](#).
- Implementar arreglos y sus operaciones utilizando POO.
- Resolver problemas utilizando arreglos.

#### 0. Implementar arreglos y sus operaciones utilizando POO:

A continuación, se presenta el archivo *main.cpp*, donde se instancia el objeto *x* de la clase *arreglo* (enviando "3" al constructor como parámetro) y se ejecutan algunos métodos de la clase:

```
1 #include<bits/stdc++.h>
2 #include"arreglo.hpp"
3
4 using namespace std;
5
6 int main()
7 {
8     arreglo x(3);
9     x.insertar_fin(1); // [1]
10    x.insertar_fin(2); // [1,2]
11    x.insertar_fin(3); // [1,2,3]
12    x.insertar_fin(4); // overflow
13    x.imprimir(); // 1 2 3
14    x.elimina_inicio(); // elimina 1
15
16    x.imprimir(); // 2 3
17    x.elimina_inicio(); // elimina 2
18    x.imprimir(); // 3
19    x.elimina_inicio(); // elimina 3
20    x.imprimir(); //
21    x.elimina_inicio(); // underflow
22    x.insertar_fin(4); // [4]
23    x.insertar_fin(5); // [4,5]
24    x.insertar_fin(6); // [4,5,6]
25    x.imprimir(); // 4 5 6
26    x.buscar(1); // 1 ENCONTRADO
27    x.buscar(6); // 6 NO encontrado
28    return 0;
}
```

Sin embargo, ¿Dónde está la clase *arreglo* y dónde están definidos sus métodos?

Podemos ver en la línea 2 (cabecera) la inclusión de un archivo *arreglo.hpp*, es en este archivo donde se ha implementado la clase *arreglo* junto con los métodos *insertar\_fin*, *elimina\_inicio*, *buscar* e *imprimir* de la clase:

```
1 #include<iostream>
2
3 using namespace std;
4
5 class arreglo
6 {
7 public:
8     int *key;
9     int max;
10    int size;
11
12    arreglo(int s)
13    {
14        key=new int[s];
15        this->max=s;
16        this->size=0;
17    }
18
19    void insertar_fin(int k);
20    void elimina_inicio();
21    void buscar(int k);
22    void imprimir();
23 };
24
25 void arreglo::insertar_fin(int k)
26 {
27     if(this->size<this->max)
28     {
29         this->key[this->size]=k;
30         this->size++;
31     }
32     else
33         cout<<"OVERFLOW"<<endl;
34 }
35
36 void arreglo::elimina_inicio()
37 {
38     if(this->size!=0)
39     {
40         for(int i=1;i<this->size;i++)
41             this->key[i-1]=this->key[i];
42         this->size--;
43     }
44     else
45         cout<<"UNDERFLOW"<<endl;
46 }
47
48 void arreglo::buscar(int k)
49 {
50     for(int i=0;i<this->size;i++)
51         if(this->key[i]==k)
52         {
53             cout<<k<<" ENCONTRADO"<<endl;
54             return;
55         }
56     cout<<k<<" NO encontrado"<<endl;
57 }
58
59 void arreglo::imprimir()
60 {
61     for(int i=0;i<this->size;i++)
62         cout<<this->key[i]<<endl;
63 }
```

Examine los métodos implementados. Compile y ejecute *main.cpp*. Analice los resultados.

**1. Implementar, en el archivo de cabecera “arreglo.hpp” y la clase arreglo, los siguientes métodos: (1 punto c/u)**

- insertar\_inicio(int k) //inserta el elemento k al inicio del arreglo.
  - elimina\_fin() //elimina el último elemento del arreglo.
  - eliminar\_key(int k) //elimina la primera aparición del elemento k.
  - imprimir\_al\_reves //imprime el arreglo de fin a inicio
  - insertar\_antes\_de\_key(int k, int kk) //inserta el elemento kk antes de la primera aparición de k, k está en lista.
  - eliminar\_antes\_de\_key(int k) // elimina el elemento anterior a key, key está en lista.
  - eliminar\_inicio\_fin() //elimina el primer elemento y el último elemento de la lista.
  - suma() //retorna la suma de todos los elementos del arreglo
  - numero\_pares() //retorna el número de elementos que son pares en el arreglo
  - invertir() //invierte el orden de los elementos en el arreglo (NO es imprimir al revés)
- \*NOTA: ninguna de las operaciones solicitadas imprime algo en pantalla.

**2. Implementar, en un nuevo archivo cabecera “arreglo\_ordenado.hpp” la clase “arreglo\_ordenado” con los siguientes métodos: (2 puntos c/u)**

- insertar(int k) //insertar un elemento k de forma ordenada en el arreglo.
- eliminar(int k) //elimina la primera aparición del elemento k en el arreglo.
- imprimir() //imprime los elementos del arreglo.
- buscar(int k) //realiza búsqueda binaria del elemento k, imprime el resultado.