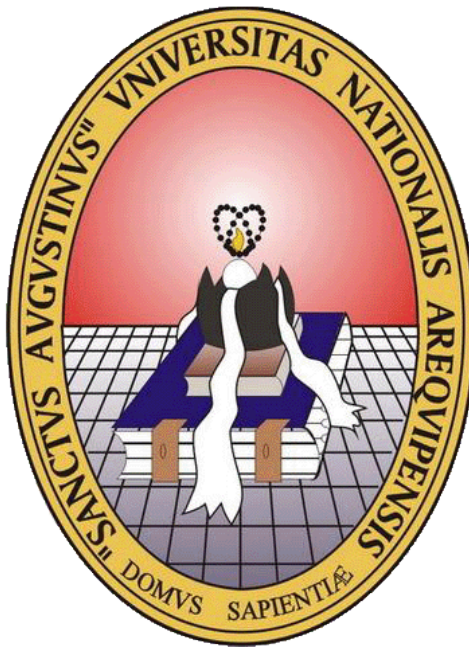


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN



PRÁCTICA DE LABORATORIO 12
CURSO DE CIENCIAS DE LA COMPUTACIÓN II

ESTUDIANTE:
RUIZ MAMANI, EDUARDO GERMÁN

EMAIL: eruizm@unsa.edu.pe

CUI: 20193061

TURNO:

C

AREQUIPA- PERÚ

2021

LINK DEL REPOSITORIO: https://github.com/EGRM23/CCII_20193061.git

1. EJERCICIO 1, 2 y 3

1. Defina una Cola que permita insertar elementos utilizando clases.
2. Sobre el ejercicio anterior, adecue el programa para eliminar elementos de una Cola.
3. Implemente un algoritmo para buscar elementos de la Cola.

- **CÓDIGO**

- **Item.h**

```
#ifndef ITEM_H
#define ITEM_H
#include <iostream>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI: 20193061

template <typename R> class Cola;
template <typename R> class Item {
public:
    Item(R val, Item* = NULL);
    ~Item();
    friend class Cola<R>;
private:
    Item* detras;
    R valor;
};

template <typename R>
Item<R> :: Item(R val, Item* d) {
    valor = val;
    detras = d;
}

template <typename R>
Item<R> :: ~Item() {}

#endif
```

- **Item.cpp**

```
#include "Item.h"

//EDUARDO GERMAN RUIZ MAMANI
//CUI: 20193061
```

- **Cola.h**

```
#ifndef COLA_H
#define COLA_H
#include <iostream>
#include "Item.h"
using namespace std;
```

```
//EDUARDO GERMAN RUIZ MAMANI  
//CUI: 20193061
```

```
template <typename R> class Item;  
template <typename R> class Cola {  
public:  
    Cola(string);  
    ~Cola();  
    void ingreso(const R val);  
    void salida();  
    R& primero();  
    string getnombre() {return nom;}  
    bool find(const R val);  
    void mostrarcola();  
private:  
    Item<R>* thefirst;  
    Item<R>* thelast;  
    int cant;  
    string nom;  
};  
  
template <typename R>  
Cola<R> :: Cola(string n) {  
    thefirst = NULL;  
    thelast = NULL;  
    cant = 0;  
    nom = n;  
}  
  
template <typename R>  
Cola<R> :: ~Cola() {  
    Item<R> *temp = thefirst;  
    Item<R> *borrar;  
    while(temp!=NULL){  
        borrar = temp;  
        temp = temp->detras;  
        delete borrar;  
    }  
}  
  
template <typename R>  
void Cola<R> :: ingreso(const R val) {  
    Item<R>* nuevo = new Item<R>(val);  
  
    if (thefirst == NULL) {  
        nuevo->detras = NULL;  
        thefirst = nuevo;  
    } else  
        thelast->detras = nuevo;  
  
    thelast = nuevo;  
    cant++;  
}  
  
template <typename R>
```

```

void Cola<R> :: salida() {
    Item<R>* temp = thefirst;

    if (thefirst->detras == NULL)
        thelast = NULL;

    thefirst = thefirst->detras;
    delete temp;
    cant--;
}

template <typename R>
R& Cola<R> :: primero() {
    return thefirst->valor;
}

template <typename R>
bool Cola<R> :: find(const R val) {
    Item<R>* temp = thefirst;

    while(temp != NULL){
        if (temp->valor == val)
            return true;
        else
            temp = temp->detras;
    }

    return false;
}

template <typename R>
void Cola<R> :: mostrarcola() {
    cout << "\nCOLA " << nom << ": \n";
    Item<R>* temp = thefirst;
    while (temp != NULL) {
        cout << temp->valor << endl;
        temp = temp->detras;
    }
}

#endif

```

- **Cola.cpp**

```
#include "Cola.h"
```

```
//EDUARDO GERMAN RUIZ MAMANI
//CUI: 20193061
```

- **main.cpp**

```
#include<iostream>
#include "Item.h"
#include "Cola.h"
using namespace std;
```

```
//EDUARDO GERMAN RUIZ MAMANI
```

```
//CUI: 20193061
```

```
int main (int argc, char *argv[]) {

    Cola<int> c1("1");
    cout << "Se ha creado la pila 1\n";

    c1.ingreso(5);
    c1.ingreso(7);
    c1.ingreso(10);
    c1.mostrarcola();
    cout << "---> Se ha ingresado 3 datos\n";

    c1.salida();
    c1.mostrarcola();
    cout << "---> Se ha eliminado 1 dato\n";

    c1.salida();
    c1.mostrarcola();
    cout << "---> Se ha eliminado 1 dato\n";

    c1.salida();
    c1.mostrarcola();
    cout << "---> Se ha eliminado 1 dato\n";

    c1.ingreso(9);
    c1.ingreso(12);
    c1.ingreso(15);
    c1.mostrarcola();
    cout << "---> Se ha ingresado 3 datos\n";

    int n = 12;
    cout << "\nBuscando " << n << "... \n";
    if (c1.find(n) == true)
        cout << "Elemento encontrado\n";
    else
        cout << "No se encuentra en la cola\n";

    n = 7;
    cout << "Buscando " << n << "... \n";
    if (c1.find(n) == true)
        cout << "Elemento encontrado\n";
    else
        cout << "No se encuentra en la cola\n";

    return 0;
}
```

- **CAPTURAS**

```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Se ha creado la pila 1

COLA 1:
5
7
10
---> Se ha ingresado 3 datos

COLA 1:
7
10
---> Se ha eliminado 1 dato

COLA 1:
10
---> Se ha eliminado 1 dato

COLA 1:
---> Se ha eliminado 1 dato

COLA 1:
9
12
15
---> Se ha ingresado 3 datos

Buscando 12...
Elemento encontrado
Buscando 7...
No se encuentra en la cola
-----
```

2. EJERCICIO 4

4. Escribir un programa que permita comparar las edades de diferentes elementos. Debe utilizar el formato de colas en clases. Se debe definir inicialmente el número de elementos y valores de cada cola (pudiendo ser de diferentes tamaños ej. 2-3). Se evaluará el resultado de la comparación de los primeros elementos de las colas, realizada en un número de iteraciones 'n', en cada iteración se debe realizar el procedimiento de inserción y eliminación de elementos (rotando los elementos definidos inicialmente)



- CÓDIGO (función COMPARACION y parte del main)

```

#include<iostream>
#include "Item.h"
#include "Stack.h"
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI: 20193061

template <typename R>
void comparacion (int iter, Cola<R>* H, Cola<R>* M) {
    Cola<R>* mayor;
    int aux;
    int cont = iter;
    while (cont > 0) {

        if ((H->primero()) >= (M->primero()))
            mayor = H;
        else
            mayor = M;

        aux = H->primero();
        H->salida();
        H->ingreso(aux);

        aux = M->primero();
        M->salida();
        M->ingreso(aux);

        cont--;
    }

    cout << "En la iteracion " << iter
          << " es mayor " << mayor->getnombre() <<
endl;
}

int main (int argc, char *argv[]) {

    cout << "-----
-----";

    cout << endl << "\nPROBLEMA COMPARACION\n";

    Cola<int> H("Los Hombres");
    Cola<int> M("Las Mujeres");

    H.ingreso(22);
    H.ingreso(12);
    H.ingreso(15);
    M.ingreso(17);
    M.ingreso(21);
    M.ingreso(18);
    M.ingreso(26);

    H.mostrarcola();

```

```

        M.mostrarcola();

        cout << endl;
        comparacion(3, &H, &M);
        comparacion(10, &H, &M);
        comparacion(100, &H, &M);
        return 0;
}

```

- **CAPTURAS**

```

PROBLEMA COMPARACION

COLA Los Hombres:
22
12
15

COLA Las Mujeres:
17
21
18
26

En la iteracion 3 es mayor Las Mujeres
En la iteracion 10 es mayor Los Hombres
En la iteracion 100 es mayor Las Mujeres

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>

```