

Guía de Practica N° 3

Colas

Código de referencia: eedd_dinamicas_lineales_1.cpp

Referencias:

LSE Colas

Recomendaciones:

- ☐ Utilizar funciones para el desarrollo.
- ☐ Probar los programas, en la máquina o con pruebas de escritorio.
- ☐ Utilizar nombres de fácil lectura para las variables.

```
struct nodoCola
{
    int dato;
    struct nodoCola* link;
};
typedef struct nodoCola NCola;
```

1. Dada una COLA vacía de nombre C del tipo NCola, cargue los valores 5, 3 y 8. Luego obtenga un valor de C y muestre por consola.

Luego mostrar el estado de la cola C

vacía={verdadero, falso}

2. Dada una COLA de nombre C del tipo NCola, vacía, realice una función que la cargue con los valores incluidos en un vector de enteros pasado como argumento. A modo de ejemplo se pide probar esta función pasando como parámetro el vector `int v[] {9,1,3};`

Luego implemente otra función que descargue completamente la cola mostrando su contenido por consola:

Retiramos: 9, 1, 3

Luego mostrar el estado de la cola C

vacía={verdadero, falso}

3. Se desea realizar la simulación de una farmacia que posee un sistema `turnador` que entrega turnos para 3 colas:

- Cola O: Clientes con Obra Social.
- Cola P: Clientes Particulares.
- Cola E: Mujeres embarazadas y jubilados, independientemente si tienen obra social.

Cada caso contendrá: el dni, el tipo de cola que usará, la duración de la atención, y el minuto en que arriba a la farmacia.

Implemente una cola de pacientes con toda su información, considerando casos donde arriben pacientes de todo tipo. Pruebe la cola ingresando y obteniendo los siguientes casos concretos:

1. Alta ArriboMinuto 0: Cliente Particular dni 12, tiempo de atención 8 minutos.
2. Alta ArriboMinuto 5: Cliente Jubilado de 77 años dni 3, tiempo atención 19 min.
3. Alta ArriboMinuto 7: Cliente Particular dni 78, tiempo de atención 4 minutos.
4. Baja: Cliente Particular dni 12.
5. Alta ArriboMinuto 10: Cliente Obra Social dni 14, tiempo de atención 11 minutos.
6. Baja: Cliente Jubilado dni 3.

Vacíe la cola, descargando completamente su contenido y mostrando por consola esta funcionalidad. Compruebe antes y después de esta operación si la cola está vacía o no.

Salida Esperada:

```
Cola de Farmacia - punto 3
-----

Arribos y bajas solicitados
-----
--> [Agregar minuto:0] (dni 12,duracion 8,tipo P)
--> [Agregar minuto:5] (dni 3,duracion 19,tipo E)
--> [Agregar minuto:7] (dni 78,duracion 4,tipo P)
--> [Obtener minuto:0] (dni 12,duracion 8,tipo P)
--> [Agregar minuto:7] (dni 78,duracion 4,tipo P)
--> [Agregar minuto:10] (dni 14,duracion 11,tipo O)
--> [Obtener minuto:5] (dni 3,duracion 19,tipo E)

Estado de la cola, vacia: falso

Vacizando la cola
-----
--> [Vaciado minuto:7] (dni 78,duracion 4,tipo P)
--> [Vaciado minuto:7] (dni 78,duracion 4,tipo P)
--> [Vaciado minuto:10] (dni 14,duracion 11,tipo O)

Estado de la cola, vacia: verdadero
```

4. Implemente una segunda versión de la cola de farmacias que permita generar tres colas con la estructura de pacientes completa, considerando casos donde arriben pacientes de todo tipo. Pruebe la versión ingresando y obteniendo los siguientes casos concretos.

```
1. Alta Minuto 0: Cliente Particular dni 12, tiempo de atención 8 minutos.
2. Alta Minuto 5: Cliente Jubilado de 77 años dni 3, tiempo de atención 19 min.
3. Alta Minuto 7: Cliente Particular dni 78, tiempo de atención 4 minutos.
4. Baja de colaP:
5. Alta Minuto 10: Cliente Obra Social dni 14, tiempo de atención 11 minutos.
6. Alta Minuto 14: Cliente Obra Social dni 35, tiempo de atención 14 minutos.
7. Alta Minuto 18: Cliente Mujer Embarazada dni 49, tiempo de atención 12.
8. Alta Minuto 20: Cliente Obra Social dni 19, tiempo de atención 20 minutos.
9. Alta Minuto 22: Cliente Obra Social dni 28, tiempo de atención 17 minutos.
10. Alta Minuto 26: Cliente Particular dni 33, tiempo de atención 2 min.
11. Alta Minuto 30: Cliente Mujer Embarazada dni 20, tiempo de atención 8 min.
12. Baja de colaE:
```

Vacíe las colas, intercalando los clientes de las tres colas en las bajas. Las colas deben quedar completamente vacías. y debe mostrar por consola la salida de esta operación. Compruebe antes y después de esta funcionalidad si las colas quedan vacías.

Salida Esperada:

```
Cola de Farmacia - punto 4 (estructura completa)
-----

Simulacion
-----
--> [A minuto: 0] (dni 12,dur 8,tipo P,estado A)
--> [A minuto: 5] (dni 03,dur 19,tipo E,estado A)
--> [A minuto: 7] (dni 78,dur 4,tipo P,estado A)
--> [B          ] (dni 12,dur 8,tipo P,estado A)
--> [A minuto:10] (dni 14,dur 11,tipo O,estado A)
--> [A minuto:14] (dni 35,dur 14,tipo O,estado A)
--> [A minuto:18] (dni 49,dur 12,tipo E,estado A)
--> [A minuto:20] (dni 19,dur 20,tipo O,estado A)
--> [A minuto:22] (dni 28,dur 17,tipo O,estado A)
--> [A minuto:26] (dni 33,dur 2,tipo P,estado A)
--> [A minuto:30] (dni 20,dur 8,tipo E,estado A)
```

```
--> [B          ] (dni 03,dur 19,tipo E,estado A)
```

```
-----  
Estado de la colaE, vacia: falso  
Estado de la colaO, vacia: falso  
Estado de la colaP, vacia: falso
```

```
Vaciando la colaE
```

```
-----  
--> [A minuto:18] (dni 49,dur 12,tipo E,estado A)  
--> [A minuto:30] (dni 20,dur 8,tipo E,estado A)  
-----
```

```
Vaciando la colaO
```

```
-----  
--> [A minuto:10] (dni 14,dur 11,tipo O,estado A)  
--> [A minuto:14] (dni 35,dur 14,tipo O,estado A)  
--> [A minuto:20] (dni 19,dur 20,tipo O,estado A)  
--> [A minuto:22] (dni 28,dur 17,tipo O,estado A)  
-----
```

```
Vaciando la colaP
```

```
-----  
--> [A minuto: 7] (dni 78,dur 4,tipo P,estado A)  
--> [A minuto:26] (dni 33,dur 2,tipo P,estado A)  
-----
```

```
Estados
```

```
-----  
Estado de la colaE, vacia: verdadero  
Estado de la colaO, vacia: verdadero  
Estado de la colaP, vacia: verdadero
```

```
estado servidor 1: Libre  
estado servidor 2: Libre  
-----
```

5. Desarrolle una simulación concreta, donde se generen los arribos siguientes con la información completa para procesarlo. Además, considere que la farmacia tiene 2 empleados para atender.

Datos de arribo:

1. Alta Minuto 0: Cliente Particular dni 12, tiempo de atención 8 minutos.
2. Alta Minuto 5: Cliente Jubilado de 77 años dni 3, tiempo de atención 19 min.
3. Alta Minuto 7: Cliente Particular dni 78, tiempo de atención 4 minutos.
4. Alta Minuto 10: Cliente Obra Social dni 14, tiempo de atención 11 minutos.
5. Alta Minuto 14: Cliente Obra Social dni 35, tiempo de atención 14 minutos.
6. Alta Minuto 18: Cliente Mujer Embarazada dni 49, tiempo de atención 12.
7. Alta Minuto 20: Cliente Obra Social dni 19, tiempo de atención 20 minutos.
8. Alta Minuto 22: Cliente Obra Social dni 28, tiempo de atención 17 minutos.
9. Alta Minuto 26: Cliente Particular dni 33, tiempo de atención 2 minutos.
10. Alta Minuto 30: Cliente Mujer Embarazada dni 20, tiempo de atención 8 min.

Salida Esperada:

Cola de Farmacia - punto D (estructura completa)

```
-----  
Simulacion
```

```
-----  
--> [A minuto: 0] (dni 12,dur 8,tipo P,estado A,emp [00,-1,-1])  
--> [C1 minuto: 0] (dni 12,dur 8,tipo P,estado C,emp 1[00,00,08])  
--> [A minuto: 5] (dni 03,dur 19,tipo E,estado A,emp [05,-1,-1])  
--> [C2 minuto: 5] (dni 03,dur 19,tipo E,estado C,emp 2[05,05,24])  
--> [A minuto: 7] (dni 78,dur 4,tipo P,estado A,emp [07,-1,-1])  
--> [F1 minuto: 8] (dni 12,dur 8,tipo P,estado F,emp 1[00,00,08])  
--> [C1 minuto: 7] (dni 78,dur 4,tipo P,estado C,emp 1[07,08,12])  
-----
```

```
--> [A minuto:10] (dni 14,dur 11,tipo O,estado A,emp [10,-1,-1])
--> [F1 minuto:12] (dni 78,dur 4,tipo P,estado F,emp 1[07,08,12])
--> [C1 minuto:10] (dni 14,dur 11,tipo O,estado C,emp 1[10,12,23])
--> [A minuto:14] (dni 35,dur 14,tipo O,estado A,emp [14,-1,-1])
--> [A minuto:18] (dni 49,dur 12,tipo E,estado A,emp [18,-1,-1])
--> [A minuto:20] (dni 19,dur 20,tipo O,estado A,emp [20,-1,-1])
--> [A minuto:22] (dni 28,dur 17,tipo O,estado A,emp [22,-1,-1])
--> [F1 minuto:23] (dni 14,dur 11,tipo O,estado F,emp 1[10,12,23])
--> [F2 minuto:24] (dni 03,dur 19,tipo E,estado F,emp 2[05,05,24])
--> [C1 minuto:18] (dni 49,dur 12,tipo E,estado C,emp 1[18,24,36])
--> [C2 minuto:14] (dni 35,dur 14,tipo O,estado C,emp 2[14,24,38])
--> [A minuto:26] (dni 33,dur 2,tipo P,estado A,emp [26,-1,-1])
--> [A minuto:30] (dni 20,dur 8,tipo E,estado A,emp [30,-1,-1])
--> [F1 minuto:36] (dni 49,dur 12,tipo E,estado F,emp 1[18,24,36])
--> [F2 minuto:38] (dni 35,dur 14,tipo O,estado F,emp 2[14,24,38])
--> [C1 minuto:38] (dni 20,dur 8,tipo E,estado C,emp 1[30,38,46])
--> [C2 minuto:38] (dni 19,dur 20,tipo O,estado C,emp 2[20,38,58])
--> [F1 minuto:46] (dni 20,dur 8,tipo E,estado F,emp 1[30,38,46])
--> [F2 minuto:58] (dni 19,dur 20,tipo O,estado F,emp 2[20,38,58])
--> [C1 minuto:58] (dni 28,dur 17,tipo O,estado C,emp 1[22,58,75])
--> [C2 minuto:58] (dni 33,dur 2,tipo P,estado C,emp 2[26,58,60])
--> [F2 minuto:60] (dni 33,dur 2,tipo P,estado F,emp 2[26,58,60])
--> [F1 minuto:75] (dni 28,dur 17,tipo O,estado F,emp 1[22,58,75])
-----
```

Estados

```
-----
Estado de la colaE, vacia: verdadero
Estado de la colaO, vacia: verdadero
Estado de la colaP, vacia: verdadero
```

```
estado servidor 1: Libre
estado servidor 2: Libre
-----
```

6. Dado el siguiente archivo de cabecera (header) complete la codificación de las funciones ColaE_vacia, ColaE_agregar y ColaE_obtener, correspondientes a la implementación de una cola estática de enteros.

```
ColaE_int.h
#define MAX 20
/*****
// Implementación de una Cola Estática de enteros
// UADER - Algoritmos y Estructuras de Datos
*****/
class ColaLLena : public exception{}
class ColaVacía : public exception{};

struct cola_estatica {
    int dato[MAX];
    int tamaño;
    int tope;
};
typedef struct cola_estatica ColaE;

void E_inicializa (ColaE &pila, int maximo) {
}

bool E_vacia(ColaE frente???) {
}

void E_agregar(ColaE &frente????, int) {
}

int E_obtener(ColaE &frente?????) {
}
```

Utilice el archivo de cabecera ColaE_int.h, para producir una nueva versión del ejercicio 2 de esta guía.

7. Dado el siguiente archivo de cabecera (header) complete la codificación de las funciones ColaG_vacia, ColaG_agregar y ColaG_obtener, correspondientes a la implementación de una cola estática genérica que permita alojar objetos primitivos (char, int, float, string) o objetos definidos por el usuario.

```
PilaEg_generica.h
#define MAX 20
/*****
// Implementación de una Pila Estática de (void*), que permita
// alojar cualquier tipo de elementos, elementos primitivos:
// char, int, float, string, e incluso estructuras y objetos
// definidos por el usuario Personas, objetos de distinto tipo.
// UADER - Algoritmos y Estructuras de Datos
*****/
#define MAX 20

using namespace std;

class PilaLlena : public exception{};
class PilaVacía : public exception{};

struct pila_estatica
{ void * pelemento[MAX];
  int tamaño;
  int tope;
};
typedef struct pila_estatica PilaEg;

void PilaEg_inicializa (PilaEg &pila, int maximo) { }

bool PilaEg_vacia (PilaEg pila) { }

void PilaEg_agregar (PilaEg &pila, void * pelemento){ }

void * PilaEg_obtener(PilaEg &pila) { }
```

Para uso con primitivo int:

```
main()

PilaEg pila;
int * pdato;

// para hacer push
pdato = new(int);
*pdato = ;
PilaEg_agregar(pila, (void *) pdato);

// para pop
pdato = (int *) PilaEg_obtener(pila);
```

Para uso con UDT:

```
struct spersona {
    int dni;
    char nombre[40];
};

main() {
    PilaEg pila;
    spersona *pdato;

    // para hacer push
    pdato = new(spersona);
    (*pdato).dni = 13241227;
    strcpy((*pdato).nombre, "Maradona Diego");
    PilaEg_agregar(pila, (void *) pdato);

    // para hacer pop
    pdato = (spersona *) PilaEg_obtener(pila);
```