ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
Universitat Rovira i Virgili

Estructures de Dades 2019-2020

Pràctica 1. Estructures lineals

Exercici 2 / 3. Cua circular

En aquest exercici continuarem amb la implementació d'estructures de dades senzilles, en particular una cua circular. Com ja sabeu, a diferència de la pila, les cues permeten guardar i consultar dades amb una política FIFO. El elements seran consultats/extrets en ordre d'inserció.

Especificació i implementació de la cua

Existeixen diverses maneres d'implementat les llistes. Les principals són dues: llista enllaçada i vector circular. En aquest cas, continuarem treballant amb estructures estàtiques i l'implementarem sobre un vector circular. A diferència de la part descrita a la teoria s'espera que l'estructura Cua contingui: vector on s'emmagatzemaran els elements, màxim nombre d'elements que la cua pot guardar (capacitat), índex de lectura i nombre actual d'elements emmagatzemats.

Per implementar la cua farem servir llenguatge C i no es poden fer servir llibreries d'estructures de dades externes. Per uniformitat, les capçaleres de les funcions han de ser exactament les següents:

```
int Crear(Cua *ac, int size);
int Encuar(Cua *ac, int elem);
int Desencuar(Cua *ac);
int Cap(Cua c, int *elem);
int EsBuida(Cua c, bool *b);
int EsPlena(Cua c, bool *b);
int Destruir(Cua *ac);
```

Els paràmetres de retorn de les funcions retornaran el resultat d'executar l'operació i error en cas que aquest s'hagi produït. Els codis de retorn que s'esperen són:

```
#define SUCCESS 0
#define ERROR_CREAR 1
#define ERROR_DESTRUIR 2
#define CUA_NO_CREADA 3
#define CUA_BUIDA 4
#define CUA_PLENA 5
```

Pels booleans heu d'utilitzar la llibreria `stdbool.h`. Els paràmetres `c` i `ac` fan referència a la Cua i a un apuntador a la Cua, respectivament. El codi s'ha d'estructurar seguint l'esquema següent:

- Declaracions de tipus i operacions a `cua.h` [INTERFACE]
- Implementació de les operacions a `cua.c` [IMPLEMENTACIÓ]
- Programa principal `test_cua.c` que comprovi que la cua funciona [CLIENT]

Aquesta uniformitat en les capçaleres i en l'estructuració del codi resultarà essencial per formar una llibreria amb totes les estructures lineals que implementem.

El programa principal `test_cua.c` serà un petit però complet joc de proves per a comprovar que efectivament l'estructura de dades funciona segons les especificacions donades.

Aplicació: generació del Triangle de Pascal

Es demana que utilitzant cues genereu el Triangle de Pascal. A cada iteració del programa, s'usarà una cua que conté els elements de la filera anterior per generar la següent.

Opcional

- Modificació de l'estructura de la cua perquè el vector que conté els elements es redimensioni quan la cua està plena.
- Implementar una funció que, donada una cua i dues posicions, giri l'ordre dels elements de la cua compresos entre les dues posicions (incloses). Per exemple, si la cua conté 10,20,30,40,50,60,70,80,90, amb primer element de la cua (el més antic) el 10, aleshores si li passem les posicions 3 i 5, la cua passarà a contenir 10,20,50,40,30,60,70,80,90. Per a la implementació, podeu utilitzar l'estructura Pila de l'exercici 1.
- Implementar una funció que, donades dues cues ordenades, retorni una nova cua que contingui les dades de les altres dues, de forma ordenada. Per exemple, si la primera cua conté 10,20,30,40,50,60, i la segona 15,20,45,75, aleshores el resultat seria una cua amb 10,15,20,20,30,40,45,50,60,75. Podeu controlar si les cues estan inicialment ordenades.

Lliurament

- Les pràctiques són individuals
- Es fa el lliurament d'un únic arxiu `PR1.2-NOM_COGNOMS.zip` que conté:
  - Els arxius principals: `cua.h`, `cua.c`, `test_cua.c` i `trianglePascal.c`
  - Un arxiu `README.txt` on s'expliquen aquells detalls d'implementació o funcionament que creieu important remarcar, i la llista detallada d'opcions incorporades (en cas que n'hi hagi)
  - Altres arxius (`*.c`, `*.h`) que facin falta per a la implementació d'opcions