

## EXAMEN PARCIAL

Assignatura: **Estructura de dades**

Data: 14 d'abril de 2016

Curs: 2015/2016

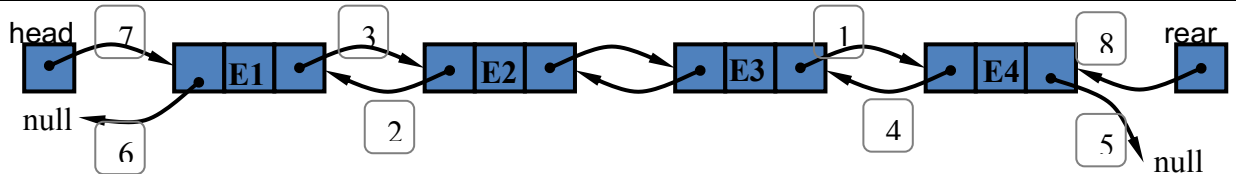


Nom: \_\_\_\_\_ Cognoms: \_\_\_\_\_ NIUB/DNI: \_\_\_\_\_  
Aula: \_\_\_\_\_ Fila: \_\_\_\_\_ Columna: \_\_\_\_\_

### Preguntes i Problemes

Temps: 90 minuts

1. (1.5p) Partint d'una cua (LinkedQueue) amb una representació amb enllaços dobles (DoubleNode) i sense capçaleres que té un punter al head i un punter al rear. Feu un swap del node del front amb el node del rear (no s'ha de fer swap de les dades). Assumiu que els nodes tenen un punter public next i un punter public prev. Justifica la teva codificació amb dibuixos, pas a pas.



```
void LinkedQueue::swapHeadRear()
{
    DoubleNode * A, *B;
    A = head->next ; // Següent del head
    B = rear->prev ; // Anterior del rear
    B->next = head;    // 1. ó B->next = A->prev; rear->prev->next = head
    A->prev = rear;    // 2. ó A->prev = B->next; head->next->prev = rear
    head->next = NULL; // 3.
    rear->prev = NULL; // 4.
    rear->next = A;    // 5.
    head->prev = B;    // 6.
    head = A->prev;    // 7.
    rear = B->next;    // 8.
}
```

2. (0.5p) Del TAD Stack, definiu la seva especificació, tot indicant quins mètodes són constructors, destructors, modificadors i consultors. Definiu l'especificació en C++ , tal i com quedaria al .h amb una representació amb arrays.

```
Class Stack{
public:
    Stack (int size); // constructor
    ~Stack() ; // Destructor
    // Consultors
    int size () const;
    bool empty() const;
    bool full () const;
    Element & top();
    // Modificadors
    void push(Element & e);
    void pop();
private:
    Element [] array ;
    int mida;
    int top_pos;
};
```

3. (2 p) Partint d'una STACK amb una representació amb enllaços simples que té un punter al top de la pila anomenat `first`. Implementeu una funció que desplaça el node que hi ha al top `n` posicions a la pila (sense usar les operacions del TAD pila), el nou top serà el node següent a l'anterior top. Assumiu que els nodes (`SingleNode`) tenen un punter public `next`. S'aconsella que useu dibuixos per justificar la vostra codificació. Quines excepcions has controlat?

```
void LinkedStack::moveTopNPositions(int n)
{
    SingleNode * aux, * temp; aux= first; temp = aux;
    if (first != null)
    {
        for (int i = 0; i< n; i++)
        {
            if (temp->next == null) throw new out_of_range("n massa gran");
            else temp = temp->next;
        }
        first = aux->next;
        aux->next = temp->next;
        temp->next = aux;
    } else throw new out_of_range("Empty Stack");
}
```

4. (0.5p) En les piles o cues amb una representació encadenadas, els nodes en quina part de la memòria RAM es guarden en temps d'execució? Justifiqueu la vostra resposta

En el HEAP !!!

5. (2p) Partint d'una `Queue<int> Q`, escriu un mètode en C++ que trobi el màxim element guardat a la cua. Només es poden usar les operacions de la cua: `enqueue`, `dequeue`, `empty`, etc. No es pot usar cap altra estructura de dades, només cues. La Queue d'entrada ha de quedar intacta després de buscar el màxim element de la cua.

```
int findMax(Queue<int> &Q)
{
    int mida = Q.size();
    int max = 0;
    if (mida == 0) throw new out_of_range("Empty Queue");
    else
    {
        int element = 0;
        for (int i = 0; i< mida; i++)
        {
            element = Q.front();
            Q.dequeue();
            max = (element > max) ? element: max;
            Q.enqueue(element);
        }
    }
    return max;
}
```