



QUADERN D'EXERCICIS DE:
ESTRUCTURES DE DADES I ALGORÍTMICA
(3105G07010/2011)

Alumne:

Charly REPULLO LAUKS

23 d'agost de 2012

Índex

| | | |
|----------|----------------------------------|-----------|
| 1 | Tutorial | 5 |
| 1.1 | Exercici 1 | 5 |
| 2 | Estructures dinàmiques I | 11 |
| 2.1 | Exercici 1 | 11 |
| 2.2 | Exercici 2 | 15 |
| 2.3 | Exercici 3 | 19 |
| 3 | Estructures dinàmiques II | 25 |
| 3.1 | Exercici 1 | 25 |
| 3.2 | Exercici 2 | 30 |
| 3.3 | Exercici 3 | 34 |
| 4 | Estructures lineals | 41 |
| 4.1 | Exercici 1 | 41 |
| A | Dates límit | 47 |

Tutorial

1.1 Exercici 1

Es disposa d'una estructura dinàmica simplement encadenada amb un punter a l'inici tal i com es representa a la figura que hi ha al final del text. Implementar en C++ el mètode `AfegirFinal` i usar-lo amb el següent programa (`main.cpp`):

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

void main() {
    estructuraDinamica e;
    int n;

    cin >> n;
    while (n != 0) {
        e.AfegirFinal(n);
        cin >> n;
    }
    e.Llistar();
}
```

Cal tenir en compte que `AfegirFinal` no verifica mai si ja existeixen els elements. El fitxer de capçalera és el següent (`estructuraDinamica.h`):

```
#ifndef TAD_estructuraDinamica_h
#define TAD_estructuraDinamica_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Estructura dinamica
    node * inici;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);
    void AfegirFinal(int i);
    void Llistar() const;
};
#endif
```

El constructor per defecte i el mètode Llistar són els següents:

```
estructuraDinamica::estructuraDinamica() {
    inici = NULL;
}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != NULL) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    cout << endl;
}
```

Cal descarregar el codi adjunt al problema. Aquest exercici conté els següents fitxers adjunts: estructuraDinamica.cpp, estructuraDinamica.h, main.cpp, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```
/*
 *  estructuraDinamica.cpp
 *  EstructuraDinamica
 *
 *  Created by Didac Barragan & Joan Surrell.
 *  Copyright (c) 2004 IMA-UdG.
 *
 *  JSS - 2008
 *  cal afegir-hi els metodes a implementar
 */

#include <cstdlib>
#include <iostream>

#include "estructuraDinamica.h"

estructuraDinamica::estructuraDinamica()
{
    inici = NULL;
}

void estructuraDinamica::Llistar() const
{
    node * p;
```

```

        p = inici;
        while (p != NULL)
        {
            cout << p->dada << " ";
            p = p-> seguent;
        }
        cout << endl;
    }

void estructuraDinamica::AfegirFinal(int i){

    node * p;
    p=inici;
    node * q = new node;
    q->dada=i;
    if(inici==NULL){
        inici=q;
        q->seguent=NULL;
        p=q;
    }
    else{
        while (p->seguent != NULL){
            p=p->seguent;
        }
        p->seguent=q;
        p=q;
    }
}

```

Nom del fitxer:estructuraDinamica.h

Codi:

```

/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Joan Surrell on Wed Oct 06 2004.
 *  Copyright (c) 2004-2006 IMA-UdG. All rights reserved.
 *
 *  Aquest fitxer es comu a tots els programes que usen l'
 *  estructura dinamica
 *  simplement encadenada. Cal implementar nomes els
 *  metodes que demana cada
 *  enunciat.
 */

#ifndef TAD_estructuraDinamica_h
#define TAD_estructuraDinamica_h

#include <cstdlib>
#include <iostream>
using namespace std;

struct node
{
    int dada;
    node *seguent;
};

```

```

class estructuraDinamica
{
    node * inici;
public:
    estructuraDinamica();
    estructuraDinamica(const estructuraDinamica & e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void InserirOrdenadament(int i);

    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);

    void Esborrar(int i);
    int TreureFinal();
    bool Existeix(int i) const;

    void OmplirInici(int n);
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar() const;

    //~estructuraDinamica();
};

#endif

Nom del fitxer:main.cpp
Codi:
#include <iostream>
#include "estructuraDinamica.h"
using namespace std;

////////// Problema 2
/*
 * Exemples d'execucio
 *
 * entrada:
 * 5 9 10 0
 * sortida:
 * 5 9 10
 *
 * entrada:
 * 9 7 5 3 1 0
 * sortida
 * 9 7 5 3 1
 */
int main()
{
    estructuraDinamica e;
    int n;

    cin >> n;

```

```
        while (n != 0) {
            e.AfegirFinal(n);
            cin >> n;
        }
        e.Llistar();

        return 0;
    }
```


Estructures dinàmiques I

2.1 Exercici 1

Es disposa d'una estructura dinàmica circular simplement encadenada amb sentinella i un punter a l'inici tal i com es representa a la figura que hi ha al final del text. Implementar en C++ els mètodes `AfegirInici` i `Existeix` que, respectivament, posen un valor a l'inici de l'estructura i retornen un booleà indicant si el valor donat existeix o no a l'estructura. Cal tenir en compte que el sentinella s'ha d'usar en totes les cerques (cerca amb sentinella) i no s'ha de modificar el seu valor a la resta d'operacions. Usar les operacions anteriors en el següent programa (`main.cpp`):

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

void main() {
    estructuraDinamica e1, e2;
    int n;

    cin >> n;
    while (n != 0) {
        if (n > 0)
            e1.AfegirInici(n);
        else if (e1.Existeix(-n))
            e2.AfegirInici(-n);
        cin >> n;
    }
    e1.Llistar();
    e2.Llistar();
}
```

Cal tenir en compte que `AfegirInici` no verifica si ja existeixen els elements. El fitxer de capçalera és el següent (`estructuraDinamica.h`):

```
#ifndef TAD_estructuraCircularSentinella_h
#define TAD_estructuraCircularSentinella_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Circular Sentinella
    node * inici;
public:
```

```

    estructuraDinamica();
    bool Existeix(int i) const;
    void AfegirInici(int i);
    void Llistar() const;
};
#endif

```

El constructor per defecte i el mètode Llistar són els següents:

```

estructuraDinamica::estructuraDinamica() {
    inici = new node; inici->seguent = inici; inici->dada = 0;
}

void estructuraDinamica::Llistar() const {
    if (inici != inici->seguent) {
        node * p = inici->seguent;
        while (p != inici) {
            cout << p->dada << " "; p = p->seguent;
        }
    }
    cout << "S:" << inici->dada << endl; // llista el sentinella
}

```

Cal descarregar el codi adjunt al problema.

Aquest exercici conté els següents fitxers adjunts: estructuraDinamica.h, estructuraDinamica.cpp, main.cpp, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```

/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Santi Espigule & Joan Surrell.
 *  Copyright (c) 2004, 2005 IMA-UdG.
 *
 *  Aquest fitxer es comu a tots els programes que usen l'
    estructura circular
 *  simplement encadenada amb sentinella.
 *
 *  JSS - 2008
 *      cal afegir-hi els metodes a implementar
 *  JSS [ACS] - 2010
 *      s'ha canviat una diferencia amb el codi de l'
    enunciat
 */

#include "estructuraDinamica.h"
#include <iostream>

```

```

using namespace std;

estructuraDinamica::estructuraDinamica()
{
    inici = new node;
    inici->seguent = inici;
    inici->dada = 0;
}

void estructuraDinamica::Llistar() const
{
    if (inici != inici->seguent) {
        node *p = inici->seguent;
        while (p != inici) {
            cout << p->dada << " ";
            p = p->seguent;
        }
        cout << "S:" << inici->dada << endl; // llista el
            sentinella
    }
}

void estructuraDinamica::AfegirInici(int i){

    node *p;
    node *q=new node;
    q->dada=i;
    p=inici->seguent;
    if(inici->seguent==inici){
        inici->seguent=q;
        q->seguent=inici;
    }
    else{
        inici->seguent=q;
        q->seguent=p;
    }
}

bool estructuraDinamica::Existeix(int i)const{

    bool b=false;
    node *p;
    p=inici->seguent;
    inici->dada=i;
    while(p!=inici && !b){
        b=p->dada==i;
        p=p->seguent;
    }
    return b;
}

Nom del fitxer:estructuraDinamica.h
Codi:
/*
 * estructuraDinamica.h
 * EstructuraDinamica
 */

```

```

* Created by Santi Espigule & Joan Surrell.
* Copyright (c) 2004, 2005 IMA-UdG.
*
* Aquest fitxer es comu a tots els programes que usen l'
  estructura circular
* simplement encadenada amb sentinella. Cal implementar
  nomes els metodes
* que demana cada enunciat.
*/

#ifndef TAD_EstructuraCircularSentinella_h
#define TAD_EstructuraCircularSentinella_h

#include <cstdlib>
#include <iostream>
using namespace std;

struct node
{
    int dada;
    node * seguent;
};

class estructuraDinamica // circular sentinella
{
    node *inici;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica &e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);
    void InserirOrdenadament(int i);

    bool Existeix(int i) const;
    int TreureFinal();
    void Esborrar(int i);

    void OmplirInici(int n);
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar( ) const;

    //~estructuraDinamica();
};

#endif

Nom del fitxer: main.cpp
Codi:
#include <iostream>
#include "estructuraDinamica.h"

```

```

using namespace std;

////////// Programa 8
/*
 * Exemples d'execucio
 *
 * entrada:
 * 3 4 2 -4 0
 * sortida:
 * 2 4 3 S:4
 * 4 S:0
 *
 * entrada:
 * 9 7 5 -5 0
 * sortida:
 * 5 7 9 S:5
 * 5 S:0
 */
int main () {
    int n;
    estructuraDinamica e1, e2;

    cin >> n;
    while (n != 0) {
        if (n > 0)
            e1.AfegirInici(n);
        else if (e1.Existeix(-n))
            e2.AfegirInici(-n);

        cin >> n;
    }
    e1.Llistar();
    e2.Llistar();

    return 0;
}

```

2.2 Exercici 2

Es disposa d'una estructura dinàmica simplement encadenada amb un punter a l'inici i un altre al final tal i com es representa a la figura que hi ha al final del text. Implementar en C++ el mètode `AfegirFinal` i usar-lo amb el següent programa (`main.cpp`):

```

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

void main() {
    estructuraDinamica e;
    int n;

    cin >> n;
    while (n != 0) {

```

```

        e.AfegirFinal(n);
        cin >> n;
    }
    e.Llistar();
}

```

Cal tenir en compte que AfegirFinal no verifica mai si ja existeixen els elements. El fitxer de capçalera és el següent (estructuraDinamica.h):

```

#ifndef TAD_estructuraDinamicaFinal_h
#define TAD_estructuraDinamicaFinal_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Final
    node * inici, * final;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);
    void AfegirFinal(int i);
    void Llistar() const;
};
#endif

```

El constructor per defecte i el mètode Llistar són els següents:

```

estructuraDinamica::estructuraDinamica() {
    inici = final = NULL;
}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    if (p != NULL) cout << p->dada << endl;
    else cout << endl;
}

```

Cal descarregar el codi adjunt al problema. Aquest exercici conté els següents fitxers adjunts: estructuraDinamica.cpp, estructuraDinamica.h, main.cpp, accedeix a l'ACME per descarregar-los.

Solucions:


```
Nom del fitxer:estructuraDinamica.cpp
Codi:
/*
 *  estructuraDinamica.cpp
 *  EstructuraDinamica
 *
 *  Created by Didac Barragan & Joan Surrell.
 *  Copyright (c) 2004 IMA-UdG.
 *
 */

#include <iostream.h>
#include "estructuraDinamica.h"

using namespace std;

estructuraDinamica::estructuraDinamica()
{
    inici = final = NULL;
}

void estructuraDinamica::Llistar() const
{
    node * p;

    p = inici;
    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    if (p != NULL)
        cout << p->dada << endl;
    else
        cout << endl;
}

void estructuraDinamica::AfegirFinal(int i){

    node *q =new node;
    q->dada=i;
    q->seguent=NULL;
    if(inici==NULL){
        inici=q;
        final=q;
    }
    else{
        final->seguent=q;
        final=final->seguent;
    }
}

Nom del fitxer:estructuraDinamica.h
Codi:
/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 */
```

```
*
*   Created by Joan Surrell & David Figuls on Wed Oct 06
*   2004.
*   Copyright (c) 2004 __MyCompanyName__. All rights
*   reserved.
*
*   IMPORTANT:
*   Aquest fitxer es comu a tots els programes que usen l'
*   estructura dinamica
*   simplement encadenada. Cal implementar nomes els
*   metodes que demana cada
*   enunciat.
*/

#ifndef TAD_estructuraDinamicaFinal_h
#define TAD_estructuraDinamicaFinal_h

#include <cstdlib>
#include <iostream>

using namespace std;

struct node
{
    int dada;
    node *seguent;
};

class estructuraDinamica // Final
{
    node * inici;
    node * final;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void InserirOrdenadament(int i);

    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);

    void Esborrar(int i);
    int TreureFinal();
    bool Existeix(int i) const;

    void OmplirInici(int n);
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar() const;
    // cal "comentar" el destructor si no s'implementa
    //~estructuraDinamica();
};
```

```

#endif

Nom del fitxer: main.cpp
Codi:
#include <iostream>
#include "estructuraDinamica.h"
using namespace std;

//////// Problema 2
/*
 * Exemples d'execucio
 *
 * entrada:
 * 5 9 10 0
 * sortida:
 * 5 9 10
 *
 * entrada:
 * 9 7 5 3 1 0
 * sortida
 * 9 7 5 3 1
 */
int main()
{
    estructuraDinamica e;
    int n;

    cin >> n;

    while (n != 0) {
        e.AfegirFinal(n);
        cin >> n;
    }
    e.Llistar();

    return 0;
}

```

2.3 Exercici 3

Es disposa d'una estructura dinàmica simplement encadenada amb un punter a l'inici i un altre al final tal i com es representa a la figura que hi ha al final del text. Implementar en C++ els mètodes `AfegirInici` i `AfegirAbans` que, respectivament, posen un valor a l'inici de l'estructura i abans d'un valor donat com a segon paràmetre. Usar-los en el següent programa (`main.cpp`):

```

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

void main() {

```

```

estructuraDinamica e;
int primer, anterior, n;

cin >> primer; anterior = primer;
e.AfegirInici(primer);
cin >> n;
while (n != 0) {
    e.AfegirAbans(n, anterior);
    anterior = n; cin >> n;
}
e.AfegirAbans(n, primer);
e.Llistar();
}

```

Cal tenir en compte que ni AfegirInici ni AfegirAbans verifiquen si ja existeixen els elements afegits. El fitxer de capçalera és el següent (estructuraDinamica.h):

```

#ifndef TAD_estructuraDinamicaFinal_h
#define TAD_estructuraDinamicaFinal_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Final
    node * inici, * final;
public:
    estructuraDinamica();
    void AfegirInici(int i);
    void AfegirAbans(int i, int ref);
    void Llistar() const;
};
#endif

```

El constructor per defecte i el mètode Llistar són els següents:

```

estructuraDinamica::estructuraDinamica() {
    inici = final = NULL;
}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    if (p != NULL) cout << p->dada << endl;
    else cout << endl;
}

```

Cal descarregar el codi adjunt al problema.

Aquest exercici conté els següents fitxers adjunts: estructuraDinamica.cpp, estructuraDinamica.h, main.cpp, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```
/*
 * estructuraDinamica.cpp
 * EstructuraDinamica
 *
 * Created by Didac Barragan & Joan Surrell.
 * Copyright (c) 2004 IMA-UdG.
 *
 */
```

```
#include <iostream.h>
#include "estructuraDinamica.h"
```

```
using namespace std;
```

```
estructuraDinamica::estructuraDinamica()
{
    inici = final = NULL;
}
```

```
void estructuraDinamica::Llistar() const
{
    node * p;

    p = inici;
    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    if (p != NULL)
        cout << p->dada << endl;
    else
        cout << endl;
}
```

```
void estructuraDinamica::AfegirInici(int i){
    node *q = new node;
    node *p;
    q->dada=i;
    if(inici==NULL){
        inici=q;
        q->seguent=NULL;
        final=q;
    }
    else{
        p=inici;
        inici=q;
        q->seguent=p;
    }
}
```

```

    }
}

void estructuraDinamica::AfegirAbans(int i,int j){

    node *p ,*r;
    p=inici;
    node *q = new node;
    q->dada=i;
    if(p->dada==j){
        inici=q;
        q->seguent=p;
    }
    else{
        r=p;
        p=p->seguent;
        while(p->dada!=j){
            r=p;
            p=p->seguent;
        }
        r->seguent=q;
        q->seguent=p;
    }
}

```

Nom del fitxer:estructuraDinamica.h

Codi:

```

/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Joan Surrell & David Figuls on Wed Oct 06
 *  2004.
 *  Copyright (c) 2004 __MyCompanyName__. All rights
 *  reserved.
 *
 *  IMPORTANT:
 *  Aquest fitxer es comu a tots els programes que usen l'
 *  estructura dinamica
 *  simplement encadenada. Cal implementar nomes els
 *  metodes que demana cada
 *  enunciat.
 */

#ifndef TAD_estructuraDinamicaFinal_h
#define TAD_estructuraDinamicaFinal_h

#include <cstdlib>
#include <iostream>

using namespace std;

struct node
{
    int dada;
    node *seguent;
};

```

```

class estructuraDinamica // Final
{
    node * inici;
    node * final;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void InserirOrdenadament(int i);

    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);

    void Esborrar(int i);
    int TreureFinal();
    bool Existeix(int i) const;

    void OmplirInici(int n);
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar() const;
    // cal "comentar" el destructor si no s'implementa
    //~estructuraDinamica();
};

#endif

Nom del fitxer: main.cpp
Codi:
#include <iostream>
#include "estructuraDinamica.h"
using namespace std;

////////// Problema 6
/*
 * Exemples d'execucio
 *
 * entrada:
 * 3 4 2 5 0
 * sortida:
 * 5 2 4 0 3
 *
 * entrada:
 * 9 7 5 1 0
 * sortida:
 * 1 5 7 0 9
 */
int main()
{
    estructuraDinamica e;
    int primer, anterior, n;

```

```
        cin >> primer;
        anterior = primer;
        e.AfegirInici(primer);

        cin >> n;
        while (n != 0) {
            e.AfegirAbans(n, anterior);
            anterior = n;
            cin >> n;
        }
        e.AfegirAbans(n, primer);
        e.Llistar();

        return 0;
    }
```


Estructures dinàmiques II

3.1 Exercici 1

Es disposa d'una estructura dinàmica circular simplement encadenada amb sentinella i un punter a l'inici tal i com es representa a la figura que hi ha al final del text. Implementar en C++ el mètode AfegirInici i el constructor de còpia (o constructor per referència). Cal tenir en compte que el sentinella s'ha d'usar en totes les cerques (cerca amb sentinella) i no s'ha de modificar el seu valor a la resta d'operacions. Usar les operacions anteriors en el següent programa (main.cpp):

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

void main() {
    estructuraDinamica e;
    int n;

    cin >> n;
    while (n != 0) {
        e.AfegirInici(n);
        cin >> n;
    }
    e.Llistar();

    estructuraDinamica e2(e);
    e2.Llistar();
}
```

Cal tenir en compte que AfegirInici no verifica mai si ja existeixen els elements repetits. El fitxer de capçalera és el següent (estructuraDinamica.h):

```
#ifndef TAD_estructuraCircularSentinella_h
#define TAD_estructuraCircularSentinella_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Circular Sentinella
    node * inici;
public:
    estructuraDinamica();
    estructuraDinamica(const estructuraDinamica & e);
```

```

        void AfegirInici(int i);
        void Llistar() const;
};
#endif

```

El constructor per defecte i el mètode Llistar són els següents:

```

estructuraDinamica::estructuraDinamica() {
    inici = new node; inici->seguent = inici; inici->dada = 0;
}

void estructuraDinamica::Llistar() const {
    if (inici != inici->seguent) {
        node * p = inici->seguent;
        while (p != inici) {
            cout << p->dada << " "; p = p->seguent;
        }
    }
    cout << "S:" << inici->dada << endl; // llista el sentinella
}

```

Cal descarregar el codi adjunt al problema.

Aquest exercici conté els següents fitxers adjunts: main.cpp, estructuraDinamica.cpp, estructuraDinamica.h, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```

/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Santi Espigule & Joan Surrell.
 *  Copyright (c) 2004, 2005 IMA-UdG.
 *
 *  Aquest fitxer es comu a tots els programes que usen l'
    estructura circular
    simplement encadenada amb sentinella.
 *
 *  JSS - 2008
 *      cal afegir-hi els metodes a implementar
 *  JSS [ACS] - 2010
 *      s'ha canviat una diferencia amb el codi de l'
    enunciat
 */

#include "estructuraDinamica.h"
#include <iostream>

using namespace std;

```

```

estructuraDinamica::estructuraDinamica()
{
    inici = new node;
    inici->seguent = inici;
    inici->dada = 0;
}

estructuraDinamica::estructuraDinamica(estructuraDinamica
&e){

    node *q;
    node *p= new node;
    q=e.inici;
    inici=p;
    p->dada=q->dada;
    q=q->seguent;
    while (q!=e.inici){
        node *r=new node;
        p->seguent=r;
        p=r;
        p->dada=q->dada;
        q=q->seguent;
    }
    p->seguent=inici;
}

void estructuraDinamica::Llistar() const
{
    if (inici != inici->seguent) {
        node *p = inici->seguent;
        while (p != inici) {
            cout << p->dada << " ";
            p = p->seguent;
        }
        cout << "S:" << inici->dada << endl; // llista el
            sentinella
    }
}

void estructuraDinamica::AfegirInici(int i){

    node *q = new node;
    q->dada=i;
    node *p;
    if (inici->seguent==inici){
        inici->seguent=q;
        q->seguent=inici;
    }
    else{
        p=inici->seguent;
        inici->seguent=q;
        q->seguent=p;
    }
}

estructuraDinamica::~estructuraDinamica(){

```

```

        node* aux;
        node* p=inici->seguent;
        while(p!=inici){
            aux=p;
            p=p->seguent;
            delete aux;
        }
        delete p;
    }
}

Nom del fitxer:estructuraDinamica.h
Codi:
/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Santi Espigule & Joan Surrell.
 *  Copyright (c) 2004, 2005 IMA-UdG.
 *
 *  Aquest fitxer es comu a tots els programes que usen l'
    estructura circular
 *  simplement encadenada amb sentinella. Cal implementar
    nomes els metodes
 *  que demana cada enunciat.
 */

#ifndef TAD_EstructuraCircularSentinella_h
#define TAD_EstructuraCircularSentinella_h

#include <cstdlib>
#include <iostream>
using namespace std;

struct node
{
    int dada;
    node * seguent;
};

class estructuraDinamica // circular sentinella
{
    node *inici;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica &e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);
    void InserirOrdenadament(int i);

    bool Existeix(int i) const;
    int TreureFinal();
    void Esborrar(int i);

```

```

        void OmplirInici(int n);
        void OmplirFinal(int n);
        int nElements() const;

        void Llistar( ) const;

        ~estructuraDinamica();
};

#endif

```

Nom del fitxer: main.cpp

Codi:

```

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

//////// Problema 1
/*
 * Exemples d'execucio
 *
 * entrada:
 * 5 6 7 8 0
 * sortida:
 * 8 7 6 5 S:0
 * 8 7 6 5 S:0
 *
 * entrada:
 * 1 3 5 7 0
 * sortida
 * 7 5 3 1 S:0
 * 7 5 3 1 S:0
 */
int main()
{
    estructuraDinamica e;
    int n;

    cin >> n;

    while (n != 0) {
        e.AfegirInici(n);
        cin >> n;
    }
    e.Llistar();

    estructuraDinamica e2(e);
    e2.Llistar();

    return 0;
}

```

3.2 Exercici 2

Es disposa d'una estructura dinàmica simplement encadenada amb sentinella final que disposa d'un punter a l'inici i un altre al final tal i com es representa a la figura que hi ha al final del text. Implementar en C++ els mètodes `nElements` que mira quants elements hi ha a l'estructura i `OmplirFinal` que posa un conjunt de valors en ordre creixent al final de l'estructura. Els valors a posar són els enters consecutius des de 1 fins al valor indicat pel paràmetre. Usar les operacions anteriors en el següent programa (`main.cpp`):

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

void main() {
    estructuraDinamica e;
    int primer, anterior, n;

    cin >> n;
    while (n != 0) {
        e.OmplirFinal(n);
        cin >> n;
    }
    e.Llistar();
    cout << "N.ELEMENTS: " << e.nElements() << endl;
}
```

Cal tenir en compte que `OmplirFinal` no verifica si ja existeixen els elements afegits. De fet, en crides successives, es repetiran els valors afegits. El fitxer de capçalera és el següent (`estructuraDinamica.h`):

```
#ifndef TAD_estructuraDinamicaSentinellaFinal_h
#define TAD_estructuraDinamicaSentinellaFinal_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Sentinella Final
    node * inici, * final;
public:
    estructuraDinamica();
    void OmplirFinal(int n);
    int nElements() const;
    void Llistar() const;
};
#endif
```

El constructor per defecte i el mètode `Llistar` són els següents:

```
estructuraDinamica::estructuraDinamica() {
    inici = final = new node;
    inici->seguent = NULL; inici->dada = 0;
```

```

}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    cout << "S:" << p->dada << endl; // llista el sentinella
}

```

Cal descarregar el codi adjunt al problema. Aquest exercici conté els següents fitxers adjunts: estructuraDinamica.cpp, main.cpp, estructuraDinamica.h, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```

/*
 * estructuraDinamica.cpp
 * EstructuraDinamica
 *
 * Created by Didac Barragan & Joan Surrell.
 * Copyright (c) 2004 IMA-UdG.
 *
 * JSS - 2009
 *   cal afegir-hi els metodes a implementar
 *   el destructor s'ha d'implementar si ho demana el
 *   professor
 */

```

```

#include <iostream>
#include "estructuraDinamica.h"

```

```

estructuraDinamica::estructuraDinamica()
{
    inici = final = new node;
    inici->seguent = NULL;
    inici->dada = 0;
}

```

```

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
}

```

```

        cout << "S:" << p->dada << endl; // llista el
            sentinella
    }

int estructuraDinamica::nElements() const{

    int c=0;
    node *p;
    p=inici;
    while (p!=final){
        c++;
        p=p->seguent;
    }
    return c;
}

void estructuraDinamica::OmplirFinal(int n){

    if (inici!=final){
        for (int i=1;i<=n;i++){
            node *q = new node;
            node *p;
            p=inici;
            while (p->seguent != final){
                p=p->seguent;
            }
            q->dada=i;
            q->seguent=p->seguent;
            p->seguent=q;
        }
    }
    else{
        for (int i=n;i>=1;i--){
            node *q = new node;
            node *p;
            p=inici;
            q->dada =i;
            inici=q;
            q->seguent=p;
        }
    }
}

estructuraDinamica::~estructuraDinamica(){

    node *aux;
    node *p=inici;
    while (p != NULL){
        aux=p;
        p=p->seguent;
        delete aux;
    }
}

Nom del fitxer:estructuraDinamica.h
Codi:
/*
 *  estructuraDinamica.h

```



```

* EstructuraDinamica
*
* Created by Joan Surrell & David Figuls on Wed Oct 06
  2004.
* Copyright (c) 2004 __MyCompanyName__. All rights
  reserved.
*
* IMPORTANT:
* Aquest fitxer es comu a tots els programes que usen l'
  estructura dinamica
* simplement encadenada. Cal implementar nomes els
  metodes que demana cada
* enunciat.
*/

#ifndef TAD_estructuraDinamicaSentinellaFinal_h
#define TAD_estructuraDinamicaSentinellaFinal_h

#include <cstdlib>
#include <iostream>
using namespace std;

struct node
{
    int dada;
    node *seguent;
};

class estructuraDinamica // Final
{
    node * inici;
    node * final;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void InserirOrdenadament(int i);

    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);

    void Esborrar(int i);
    bool Existeix(int i) const;

    void OmplirInici(int n);
    int TreureFinal();
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar() const;
    ~estructuraDinamica();
};

```

```

#endif

Nom del fitxer: main.cpp
Codi:
#include <iostream>
#include "estructuraDinamica.h"
using namespace std;

////////// Problema 10
/*
 * Exemples d'execucio
 *
 * entrada:
 * 1 2 3 0
 * sortida:
 * 1 1 2 1 2 3
 * N.ELEMENTS: 6
 *
 * entrada:
 * 3 2 1 0
 * sortida:
 * 1 2 3 1 2 1
 * N.ELEMENTS: 6
 */
int main()
{
    estructuraDinamica e;
    int n;

    cin >> n;
    while (n != 0) {
        e.OmplirFinal(n);
        cin >> n;
    }
    e.Llistar();
    cout << "N.ELEMENTS: " << e.nElements() << endl;

    return 0;
}

```

3.3 Exercici 3

Es disposa d'una estructura dinàmica simplement encadenada amb sentinella final que disposa d'un punter a l'inici i un altre al final tal i com es representa a la figura que hi ha al final del text. Implementar en C++ els mètodes `AfegirInici` i `Esborrar` que, respectivament, posen un valor a l'inici de l'estructura i esborren un valor donat (si existeix). Cal tenir en compte que el sentinella s'ha d'usar en totes les cerques (cerca amb sentinella) i no s'ha de modificar el seu valor a la resta d'operacions. Usar les operacions anteriors en el següent programa (`main.cpp`):

```

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

```

```

void main() {
    estructuraDinamica e;
    int n;

    cin >> n;
    while (n != 0) {
        if (n > 0)
            e.AfegirInici(n);
        else
            e.Esborrar(-n);
        cin >> n;
    }
    e.Llistar();
}

```

Cal tenir en compte que AfegirInici no verifica si ja existeixen els elements mentre que Esborrar no fa res en cas que el valor no existeixi. El fitxer de capçalera és el següent (estructuraDinamica.h):

```

#ifndef TAD_estructuraDinamicaSentinellaFinal_h
#define TAD_estructuraDinamicaSentinellaFinal_h

struct node {
    int dada;
    node *seguent;
};

class estructuraDinamica { // Sentinella Final
    node * inici, * final;
public:
    estructuraDinamica();
    void AfegirInici(int i);
    void Esborrar(int i);
    void Llistar() const;
};
#endif

```

El constructor per defecte i el mètode Llistar són els següents:

```

estructuraDinamica::estructuraDinamica() {
    inici = final = new node;
    inici->seguent = NULL; inici->dada = 0;
}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    cout << "S:" << p->dada << endl; // llista el sentinella
}

```

Cal descarregar el codi adjunt al problema. Aquest exercici conté els següents fitxers adjunts: main.cpp, estructuraDinamica.h, estructuraDinamica.cpp, accedeix a l'ACME per descarregar-los.

Solucions:

Nom del fitxer: estructuraDinamica.cpp

Codi:

```
/*
 *  estructuraDinamica.cpp
 *  EstructuraDinamica
 *
 *  Created by Didac Barragan & Joan Surrell.
 *  Copyright (c) 2004 IMA-UdG.
 *
 *  JSS - 2009
 *  cal afegir-hi els metodes a implementar
 *  el destructor s'ha d'implementar si ho demana el
 *  professor
 */

#include <iostream>
#include "estructuraDinamica.h"

estructuraDinamica::estructuraDinamica()
{
    inici = final = new node;
    inici->seguent = NULL;
    inici->dada = 0;
}

void estructuraDinamica::Llistar() const {
    node * p = inici;

    while (p != final) {
        cout << p->dada << " ";
        p = p->seguent;
    }
    cout << "S:" << p->dada << endl; // llista el
    sentinella
}

void estructuraDinamica::AfegirInici(int i){
    node *p = new node;
    p->dada=i;
    p->seguent=inici;
    inici=p;
}

void estructuraDinamica::Esborrar(int i){
```

```

    node *p=inici;
    node *q;
    final->dada=i;
    while (p->dada!=final->dada){
        q=p;
        p=p->seguent;
    }
    if (p!=final){
        if(p==inici){
            inici=p->seguent;
            delete p;
        }
        else {
            q->seguent=p->seguent;
            delete p;
        }
    }
}

estructuraDinamica::~estructuraDinamica(){
    node *aux;
    node *p=inici;
    while (p!=NULL){
        aux=p;
        p=p->seguent;
        delete aux;
    }
}

Nom del fitxer:estructuraDinamica.h
Codi:
/*
 *  estructuraDinamica.h
 *  EstructuraDinamica
 *
 *  Created by Joan Surrell & David Figuls on Wed Oct 06
 2004.
 *  Copyright (c) 2004 __MyCompanyName__. All rights
reserved.
 *
 *  IMPORTANT:
 *  Aquest fitxer es comu a tots els programes que usen l'
estructura dinamica
 *  simplement encadenada. Cal implementar nomes els
metodes que demana cada
 *  enunciat.
 */

#ifdef TAD_estructuraDinamicaSentinellaFinal_h
#define TAD_estructuraDinamicaSentinellaFinal_h

#include <cstdlib>
#include <iostream>
using namespace std;

```

```

struct node
{
    int dada;
    node *seguent;
};

class estructuraDinamica // Final
{
    node * inici;
    node * final;
public:
    estructuraDinamica();
    estructuraDinamica(estructuraDinamica & e);

    void AfegirInici(int i);
    void AfegirFinal(int i);
    void InserirOrdenadament(int i);

    void AfegirDespres(int i, int j);
    void AfegirAbans(int i, int j);

    void Esborrar(int i);
    bool Existeix(int i) const;

    void OmplirInici(int n);
    int TreureFinal();
    void OmplirFinal(int n);
    int nElements() const;

    void Llistar() const;

    ~estructuraDinamica();
};

```

```

#endif

```

Nom del fitxer: main.cpp

Codi:

```

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

```

```

//////// Programa 7

```

```

/*
 * Exemples d'execucio
 *
 * entrada:
 * 3 4 2 -4 0
 * sortida:
 * 2 3 S:4
 *
 * entrada:
 * 9 7 5 -5 0
 * sortida:
 * 7 9 S:5
 */

```

```
int main () {
    int n;
    estructuraDinamica e;

    cin >> n;
    while (n != 0) {
        if (n > 0)
            e.AfegirInici(n);
        else
            e.Esborrar(-n);

        cin >> n;
    }
    e.Llistar();
    return 0;
}
```


Estructures lineals

4.1 Exercici 1

Es disposa d'una pila de caràcters amb representació dinàmica (estructura simplement encadenada amb un punter a l'inici) tal i com es mostra a la figura que hi ha al final del text. Implementar en C++ els mètodes de la pila (constructor per defecte, Buida, Empila, Desempila i Cim).

Crear un algorisme que llegeixi una seqüència que conté només els caràcters (,), [,], {, } i acaba per punt. Cal que l'algorisme digui si la seqüència està o no ben parenteritzada (p.e.: la seqüència { () [] } està ben parenteritzada mentre que { ([]) } no ho està).

En acabar la seqüència cal escriure SI o NO segons es tracti d'una seqüència correcta o no, respectivament.

El fitxer de capçalera és el següent (pilaDinamica.h):

```
#ifndef TAD_pilaDinamica_h
#define TAD_pilaDinamica_h

struct node {
    char dada;
    node *seguent;
};

class pilaDinamica { // pila dinamica
    node * inici;
public:
    pilaDinamica();
    bool Buida() const;
    void Empila(int i);
    void Desempila();
    int Cim() const;
};
#endif
```

Un esquelet de programa principal seria (main.cpp):

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

int main() {
    pilaDinamica p;
    char c;
    bool OK = true;
```

```

    cin >> c;
    while (OK && (c != '.')) {
        // processar valor
        cin >> c;
    }

    if (OK && p.Buida())
        cout << "SI" << endl;
    else
        cout << "NO" << endl;
}

```

Exemples de funcionament:

Entrada: {[()]}.
Sortida: SI

Entrada: {{{}}}.
Sortida: NO

Entrada: {[()]}).
Sortida: NO

Solucions:

Nom del fitxer: main.cpp

Codi:

```

#include <iostream>
#include <vector>
#include "pilaDinamica.h"

```

```
using namespace std;
```

```

int main(){

    pilaDinamica p;
    char c;
    bool OK = true;
    vector<char> vc;
    //cout << "Entra primer caracter"<<endl;
    cin >> c;
    while (c!='.') {
        vc.push_back(c);
        //cout<<"inserir en vector"<<endl;
        //cout<<"Entra un altre caracter"<<endl;
        cin >> c;
    }
    OK=p.correcte(vc);
    if (OK && p.Buida()) cout << "SI" << endl;
    else cout << "NO" << endl;
}

```

```

        return 0;
    }

    Nom del fitxer:pilaDinamica.cpp
    Codi:
    #include "pilaDinamica.h"
    #include <vector>

    pilaDinamica::pilaDinamica(){
        inici=NULL;
    }

    pilaDinamica::~~pilaDinamica(){
        node *p,*q;
        p=inici;
        q=inici;
        while(p!=NULL){
            p=p->seguent;
            delete q;
            q=p;
        }
    }

    bool pilaDinamica::Buida() const{
        return inici==NULL;
    }

    void pilaDinamica::Empila(char c){
        node *p = new node;
        p->dada=c;
        p->seguent=inici;
        inici=p;
    }

    void pilaDinamica::Desempila(){
        node *p;
        p=inici;
        inici=inici->seguent;
        delete p;
    }

    char pilaDinamica::Cim() const{
        return inici->dada;
    }

    bool pilaDinamica::correcte(vector<char> v){

        bool Acabar=false;
        bool Correcte=true;
        char c;
        vector<char>::iterator it=v.begin();
        Empila('*');
        while (it!=v.end() && !Acabar){
            if(*it=='(' || *it=='[' || *it=='{'){
                c=*it;
                Empila(c);
                it++;
            }
        }
    }

```

```

    }
    else if(*it==')'){
        if(Cim()== '('){
            Desempila();
            it++;
        }
        else{
            Acabar=true;
            Correcte=false;
        }
    }

    else if(*it==']'){
        if(Cim()== '['){
            Desempila();
            it++;
        }
        else{
            Acabar=true;
            Correcte=false;
        }
    }

    else if(*it=='}'){
        if(Cim()== '{'){
            Desempila();
            it++;
        }
        else{
            Acabar=true;
            Correcte=false;
        }
    }

    }
    if(Cim()!='*')Correcte=false;
    else if(Cim()=='*')Desempila();
    return Correcte;
}

```

Nom del fitxer:pilaDinamica.h

Codi:

```

#ifndef PILADINAMICA_H
#define PILADINAMICA_H

```

```

#include <cstdlib>
#include <iostream>
#include <vector>

```

```
using namespace std;
```

```

struct node{
    char dada;
    node *seguent;
};

```

```

class pilaDinamica
{
    node * inici;

```

```
public:
    pilaDinamica();
    //pilaDinamica(const pilaDinamica & p); // no es
        demana
    void Empila(char c);
    void Desempila();
    char Cim() const;
    bool Buida() const;
    bool correcte(vector<char>);

    ~pilaDinamica(); // cal implementar-lo
protected:
private:
};

#endif // PILADINAMICA_H
```


Dates límit

- Tutorial : 9/10/2011 23:59:00
- Estructures dinàmiques I : 10/10/2011 23:59:00
- Estructures dinàmiques II : 20/10/2011 23:59:00
- Estructures lineals : 31/10/2011 23:59:00