ListaLonja.pdf

Anónimo

Fundamentos de la Programación

1º Grado en Ingeniería Informática

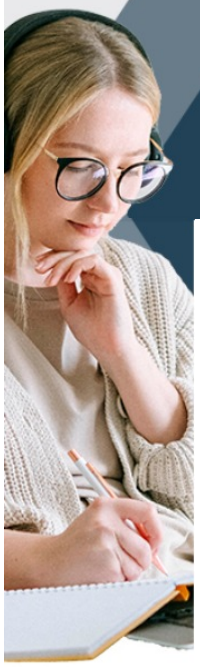
Facultad de Informática
Universidad Complutense de Madrid

Maths informática

CURSOS INTENSIVOS PARA EXÁMENES DE
CONVOCATORIA ORDINARIA
Y EXTRAORDINARIA

Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

CURSOS INTENSIVOS PARA EXÁMENES DE
CONVOCATORIA ORDINARIA
Y EXTRAORDINARIA



Academia especializada en estudios
de la Facultad de Informática

Maths informática

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

void iniziar(ListaLotes &listas) {
    listas.count = 0;
}
int numLotes(const ListaLotes &lista) {
    return lista.count;
}
void insertar(ListaLotes &listas, tLotes lote) {
    bool found = false;
    int pos;
    pos = Buscar(listas, 0, listas.count - 1, found, lote.identificacion, lote.tipo);
    if (found) {
        cout << "Lote ya esta en la lista " << endl;
    }
    else {
        int i = 0;

        while (i < listas.count && *listas.lista[i] < lote) {
            i++;
        }
        for (int j = listas.count; j > i;  j--) {
            listas.lista[j] = listas.lista[j - 1];
        }
        listas.lista[i] = new tLotes(lote);
        listas.count++;
    }
}
bool cargar(ListaLotes &listas) {
    ifstream file;
    tLotes lote;
    bool open = false;
    file.open(FILENAME);
    if (file.is_open()) {
        open = true;
        iniziar(listas);
        while (!file.eof() && numLotes(listas) < MAX) {
            file >> lote.identificacion >> lote.tipo >> lote.peso >> lote.precio;
            insertar(listas, lote);
        }

        file.close();
    }
    else {
        cout << "lfdkjs" << endl;
    }

    return open;
}
int Buscar(const ListaLotes &listas, int beg, int end, bool &found, string identificacion,
    int pos = -1, middle;
    tLotes lote;
    lote.identificacion = identificacion;
    lote.tipo = tipo;
    if (beg <= end && !found) {
        middle = (beg + end) / 2;
        if (*listas.lista[middle] == lote) {
```

Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

```
61:             found = true;
62:             pos = middle;
63:         }
64:         else if (*listas.lista[middle] < lote) {
65:             pos = Buscar(listas, middle + 1, end, found, identificacion, tipo);
66:
67:         }
68:         else {
69:             pos= Buscar(listas, beg, middle -1, found, identificacion, tipo);
70:         }
71:     }
72:
73:     return pos;
74: }
75: tLotes buscarlote(const ListaLotes &listas, int pos) {
76:     return *listas.lista[pos];
77: }
78: void mostrar(const ListaLotes &listas) {
79:     for (int i = 0; i < listas.count; i++) {
80:         mostrar(buscarlote(listas, i));
81:     }
82: }
83: void liberar(ListaLotes &listas) {
84:     for (int i = 0; i < listas.count; i++) {
85:         delete listas.lista[i];
86:     }
87: }
```

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.