

REPARACIONES.TXT - ROXANA LEITUZ 2024 VERANO (PRE-PARCIAL)

En un negocio de reparaciones se solicita un desarrollo para administrar la gestión de los trabajos.

Cada día se reciben diferentes tipos de equipos:

electrodomésticos pequeños - código 1
electrodomésticos grandes- código 2
computadoras- código 3
impresoras-código 4
sin caracterizar-código 5

Cuando se ingresa el producto se le entrega un numero de trabajo que es secuencia. Comienza en 1 y siempre se les da el siguiente del último cuyo número se encuentra en un archivo que debe ser actualizado cada vez que se lee, con el número más 1.

Los datos que se solicitan son nombreyapellido, código de equipo, fecha (día y mes, código de empleado (número entre 1 y 50), importe esperado, fecha probable de entrega (día y mes), importe a cobrar siempre en 0 cuando se da de alta.

Se solicita...

-una función que cargue los datos en una estructura que considere adecuada, ya que se debe realizar lo siguiente:

-Una función que busqué un número de trabajo para poder imprimir los datos completos del trabajo y el nombre del empleado.

- Una función void para actualizar la fecha de entrega con la real y el importe a cobrar real dado el número de trabajo, el importe y la fecha del día.

- Una función que liste los trabajos pendientes de entrega ordenados por mes, y día imprimiendo número de trabajo y nombre de empleado. Para lo cual cuenta con un archivo desordenado con los nombres y los códigos (están todos).

TXT- Practica Lotuz

c/ día se reciben tipos de equipos

electro chico - cod 1
electro grande - cod 2
compu - cod 3
impresora - cod 4
sin caracterizar - cod 5

numero-de-trabajo

↳ en un archivo.txt

```
int cant=0;
int cant=fTell(arch);
```

datos solicitados: char nombre[50];
int codequipo;
int fecha; → ddmm
int codemp;
int importe; → 0 para dar de baja
int entrega; → ddmm

quiero: función y/ cargar datos.

- ↳ busque num.de-trabajo para imprimir toda la struct!
- ↳ Actualizar la fecha de entrega con la real y el importe real dado numero-de-trabajo, importe, fecha del día
- ↳ Listar trabajos pendientes de entrega x mes y día ordenado imprimiendo num-trabajo y nom
- ↳ para esto se cuenta con un archivo designado con los nombres y códigos.

struct Fecha {

int dia;
int mes;

};

struct gestion {

int codigo_equipo;
char nombre[50];
Fecha fecha;
int codigo_emp; (1-50)
int importe;
int entrega; ddmm
int num_trabajo;

};

struct Nodo {

gestion dato;
Nodo *sigte;

};

elijo una estructura de lista!
como no se repiten los números de trabajo!

busco el insertar Ordenado con dato, num-trabajo

CargarDatos();

busquedaNT();

actualizarExivo (Nodo *d, gestion info);

listarPendentes();

Notas:

```

- void cargaDatos (Nodo * &lista, FILE * nums) {
    nums = fopen("numeros_trabajo.txt", "rb");
    int aux;
    gestion info;
    fseek(nums, 0, SEEK_SET);
    while (!feof(nums)) {
        fread(&aux, sizeof(int), 1, nums);
        info.num_trabajo = aux;
        cin >> codigo_equipo;
        cin >> nombre;
        cin >> codigo_emp;
        cin >> importe;
        cin >> entrega;
        cin >> info.fecha.dia;
        cin >> info.fecha.mes;
        insertarOrdenado(lista, info);
    }
    nums.close();
}

```

```

- actualizarEnvio (Nodo * &lista, gestion info) {
    reemplazo = buscar(lista, info);
    eliminar(lista, info);

    cin >> reemplazo.entrega;
    cin >> reemplazo.importe;
    insertarOrdenado(lista, reemplazo);
}

```

```

- busqueda Num (Nodo * lista, gestion info) {
    cin >> info.num_trabajo;
    buscado = buscar(lista, info);
    cout << nombre << " " << codigo_emp << endl << codigo_equipo << " " <<
    buscado.importe << endl << buscado.fecha.dia << "/" << buscado.fecha.mes << endl <<
    buscado.entrega << endl;
}

```

Hoja:

Fecha:

```
- void ListarPendientes (Nodo * lista) {  
  Nodo * aux = lista;  
  Nodo * pendientes = NULL;
```

```
  while (aux != NULL) {
```

```
    if ((info.fecha.dia + (100 * info.fecha.mes) > info.entrega) {  
      insertarOrdenado(pendientes);
```

```
    }
```

```
    aux = aux->pendientes;
```

```
  while (aux->info != NULL) {
```

```
    cout << num.trabajo << endl << nombre << endl;  
    aux = aux->sigte;
```

```
  }
```

```
}
```

```
if ((info.fecha.dia + (100 * info.fecha.mes) > info.entrega) {
```

```
  cout << num.trabajo << endl << nombre << endl;  
  insertarOrdenado(pendientes, aux->info);
```