Práctica 5

Listas de Registros

1. Ejecute el programa ***Dev-C++***.
2. En el menú ***Archivo*** seleccione la opción ***Nuevo*** y a continuación ***Archivo Fuente***.
3. En el nuevo documento se pide al alumno que copie esta variante del programa anterior en la que se cambia la estructura de los nodos para que además de un número incluyan el nombre de una persona y un valor numérico correspondiente a su sueldo, por simplicidad, hemos omitido el uso de plantillas en esta práctica.

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

class Nodo {
  public:
    Nodo *sig;
    int valor;
    string nombre;
    float sueldo;
    Nodo(int v, string n, float s) {
      sig=NULL;
      valor = v;
      nombre=n;
      sueldo=s;
    }
};

class Lista {
  public:
    Lista() { le = NULL; }
    ~Lista();
    void Insertar(int v,string n, float s);
    int Eliminar(int v);
    bool Vacia() { return le == NULL; }
    void Ver();
  private:
    Nodo *le;
};

Lista::~Lista() {
  Nodo *aux;
```

```
  while(le) {
    aux = le;
    le = le->sig;
    delete aux;
  }
}

void Lista::Insertar(int v,string n, float p) {
  Nodo *tmp;
  if(Vacia() || le->valor > v) {
    Nodo *nvo= new Nodo(v,n,p);
    nvo->sig=le;
    le=nvo;
  }
  else {
    tmp= le;
    while(tmp->sig && tmp->sig->valor <= v) tmp= tmp->sig;
    Nodo *nvo= new Nodo(v,n,p);
    nvo->sig=tmp->sig;
    tmp->sig = nvo;
  }
  Ver();
}

int Lista::Eliminar(int v) {
  Nodo *tmp, *nodo;
  nodo = le;
  tmp = NULL;
  while(nodo && nodo->valor < v) {
    tmp = nodo;
    nodo = nodo->sig;
  }
  if(!nodo || nodo->valor != v) return 0;
  else {
    if(nodo==le) le = nodo->sig;
    else tmp->sig = nodo->sig;
    delete nodo;
  }
  return v;
}

void Lista::Ver() {
  Nodo *tmp=le;
  if(Vacia()) cout << "La Lista esta Vacia."<<endl;
  else {
    cout <<endl<< "La Lista actual es:\n";
```

```
        while(tmp) {
          cout<< setw(5) << tmp->valor<< setw(18) <<tmp->nombre<< setw(12) <<tmp->sueldo<<endl;
          tmp= tmp->sig;
         }
      }
      cout << endl;
    }

    int main() {
      Lista le;
      string nombre;
      float sueldo;
      int valor;
      bool band;
      char opc;
      cout<<endl<<"Implementacion de una Lista con nodos complejos\n\n";
      do {
         cout<<"\n1.- Insertar \n2.- Eliminar \n3.- Ver lista actual\n4.- Salir\n\n";
         cout<<"Opcion: ";
         cin>>opc;
         switch(opc) {
            case '1': cout<<endl<<"Introduce un valor: ";
                      cin>>valor;
                      cout<<endl<<"Introduce un nombre: ";
                      cin>>nombre;
                      cout<<endl<<"Introduce un sueldo: ";
                      cin>>sueldo;
                      le.Insertar(valor,nombre,sueldo);
                      break;
            case '2': cout<<endl<<"Introduce el valor a eliminar: ";
                      cin>>valor;
                      band=le.Eliminar(valor);
                      if(!band) cout<<endl<<"El dato no se encuentra."<<endl;
                      else cout<<endl<<"El dato ha sido eliminado." <<endl;
                      le.Ver();
                      break;
            case '3': le.Ver();
                      break;
         }
      }while(opc!='4');
    }
```

4. Ahora localice el icono de ***Compilar*** que está al inicio de la barra de herramientas y oprímalo, asigne el nombre de ***ejercicio 5A*** y elija el destino que guste para guardarlo.
5. Ahora ejecútelo e inserte los valores mostrados en la siguiente tabla:

Número	Nombre	Sueldo
7	Enrique	3500
5	Jaime	14250
8	Marcos	12000
4	Pedro	2100

6. Verifique que la lista se genera de forma ordenada, pruebe que la eliminación de nodos funciona correctamente.
7. El siguiente ejercicio funciona de manera semejante, pero en este caso hemos definido una estructura llamada Persona para manipular los datos:

```
#include <iostream>
#include <iomanip>
using namespace std;

struct Persona {
  int  Valor;
  char Nombre[15];
  float Sueldo;
};

class Nodo {
   public: Persona F;
           Nodo *sig;
           Nodo(Persona);
};

class Lista {
   private: Nodo *le;
   public:  Lista();
            ~Lista();
            void Insertar(Persona);
            int Eliminar(Persona);
            void Ver();
            int Vacia();
 };

Nodo::Nodo(Persona P) {
   F = P;
   sig = NULL;
}

Lista::Lista() {
   le = NULL;
}

Lista::~Lista() {
   Nodo *sale;
```

```
    int val;
    while(le) {
        sale=le;
        le=le->sig;
        val=sale->F.Valor;
        delete(sale);
        cout<<"Bloque de memoria liberado: "<<val<<endl;
    }
    delete le;
}

void Lista::Insertar(Persona P) {
    Nodo *tmp;
    if (Vacia() || le->F.Valor > P.Valor) {
        Nodo *nvo = new Nodo(P);
        nvo->sig = le;
        le = nvo;
    }
    else {
        tmp = le;
        while(tmp->sig && tmp->sig->F.Valor <= P.Valor)  tmp = tmp->sig;
        Nodo *nvo = new Nodo(P);
        nvo->sig = tmp->sig;
        tmp->sig = nvo;
    }
    cout<<"Hemos insertado el valor. "<<P.Valor<<endl;
    Ver();
}

int Lista::Eliminar(Persona P) {
    Nodo *tmp, *nodo;
    nodo = le;
    tmp = NULL;
    while(nodo && nodo->F.Valor < P.Valor) {
        tmp = nodo;
        nodo = nodo->sig;
    }
    if(!nodo || nodo->F.Valor != P.Valor) return 0;
    else {
        if(nodo==le) le = nodo->sig;
        else tmp->sig = nodo->sig;
        delete nodo;
    }
    return P.Valor;
}
```

```
int Lista::Vacia() {
   return le==NULL ? 1 : 0;
}

void Lista::Ver() {
   Nodo *tmp=le;
   if(Vacia()) cout << "La lista esta Vacia.";
   else {
      cout << "La lista actual es: "<<endl;
      while(tmp) {
        cout << setw(5) << tmp->F.Valor<< setw(18) << tmp->F.Nombre << setw(12);
        cout << tmp->F.Sueldo<<endl;
        tmp = tmp->sig;
      }
   }
   cout << endl;
 }

int main() {
   Lista le;
   Persona Fulanito;
   int band;
   char opc;
   cout<<endl<<"Implementacion de una lista de registros\n\n";
   do {
      cout<<"\n1.- Insertar \n2.- Eliminar \n3.- Ver lista actual\n4.- Salir\n\n";
      cout<<"Opcion: ";
      cin>>opc;
      switch(opc) {
         case '1': cout<<endl<<"Introduce el Valor: ";
                   cin>>Fulanito.Valor;
                   cout<<endl<<"Introduce el nombre: ";
                   cin>>Fulanito.Nombre;
                   cout<<endl<<"Introduce el Sueldo: ";
                   cin>>Fulanito.Sueldo;
                   le.Insertar(Fulanito);
                   break;
         case '2': cout<<endl<<"Introduce un valor a eliminar: ";
                   cin>>Fulanito.Valor;
                   band=le.Eliminar(Fulanito);
                   if(band==0) cout<<endl<<"El dato no se encuentra."<<endl;
                   else cout<<endl<<"El dato ha sido eliminado."<<endl;
                   le.Ver();
                   break;
         case '3': le.Ver();
                   break;
```

```
        }
    } while(opc!='4');
}
```

8. Ahora localice el icono de ***Compilar*** que está al inicio de la barra de herramientas y oprímalo, asigne el nombre de ***ejercicio 5B*** y elija el destino que guste para guardarlo.
9. Ahora ejecútelo e inserte los valores mostrados en la tabla anterior:
10. Fin de la Práctica.