```java
package Arboles;

import java.util.Scanner;

public class ABinario {
	private Nodo raiz;
	ABinario()
	{
		raiz = null;
	}
	public Nodo getRaiz() {
		return raiz;
	}
	public void setRaiz(Nodo raiz) {
		this.raiz = raiz;
	}

	void crear(Nodo r)
	{
		if (r != null)
		{
			Scanner in = new Scanner(System.in);
			int num = in.nextInt();
			r.setNum(num);
			System.out.println("Tendra Izq? s/n");
			String res = in.next();
			if (res.equals("s"))
			{
				Nodo nuevo = new Nodo();
				r.setIzq(nuevo);
				crear(nuevo);
			}

			System.out.println("Tendra der? s/n");
			res = in.next();
			if (res.equals("s"))
			{
				Nodo nuevo = new Nodo();
				r.setDer(nuevo);
				crear(nuevo);
			}
		}
	}
	void preorder(Nodo r)
	{
		if (r != null)
		{
			System.out.println(r.getNum());
			preorder(r.getIzq());
			preorder(r.getDer());
		}
	}
	void niveles()
	{
		CSimpleD nivel = new CSimpleD();
		CSimpleD desc = new CSimpleD();
		nivel.adicionar(getRaiz());
		int c = 0;
		while (!nivel.esvacia())
		{
			System.out.println("Nivel " + c + ": ");
			while (!nivel.esvacia())
			{
				NodoAD x = nivel.eliminar();

				if (x.getIzq() != null)
					desc.adicionar(x.getIzq());
				if (x.getDer() != null)
					desc.adicionar(x.getDer());
			}
			c++;
			nivel.vaciar(desc);
		}
	}
}
```

## ARBOL

```java
package Arboles;

public class Nodo {
	private int num;
	private Nodo izq, der;
	Nodo()
	{
		izq = der = null;
	}
	public int getNum() {
		return num;
	}

	public void setNum(int num) {
		this.num = num;
	}

	public Nodo getIzq() {
		return izq;
	}

	public void setIzq(Nodo izq) {
		this.izq = izq;
	}

	public Nodo getDer() {
		return der;
	}

	public void setDer(Nodo der) {
		this.der = der;
	}


}
```

```java
package Arboles;

public class main {

	public static void main(String[] args)
{
		ABinario A = new ABinario();
		Nodo w = new Nodo();
		A.setRaiz(w);
		A.crear(A.getRaiz());
		A.mostrar(A.getRaiz());
	}

}
```

**COLACircular**

```
import java.util.Scanner;

public class ColaCircular {
	private int max = 50, ini, fin;
	private int v[] = new int [max + 1];

	ColaCircular()
	{
		ini = fin = 0;
	}
	boolean esvacia()
	{
		return nroElem() == 0;
	}
	boolean esllena()
	{
		return nroElem() == max - 1;
	}
	int nroElem()
	{
		return ((max + fin - ini) % max);
	}
	void adicionar(int x)
	{
		if (esllena())
			System.out.println("Cola LLena");
		else
		{
			fin = (fin + 1) % max;
			v[fin] = x;
		}
	}
	int eliminar()
	{
		int dat = -1;
		if (esvacia())
			System.out.println("Cola Vacia");
		else
		{
			ini = (ini + 1) % max;
			dat = v[ini];
			if (nroElem() == 0)
				ini = fin = 0;
		}
		return dat;
	}
	void llenar(int n)
	{
		Scanner in = new Scanner(System.in);
		for (int i = 0; i < n; i++)
		{
			int dat = in.nextInt();
			adicionar(dat);
		}
	}
	void mostrar()
	{
		ColaCircular aux = new ColaCircular();
		System.out.println("Datos Cola Circular:");
		while (!esvacia())
		{
			int dat = eliminar();
			System.out.print(dat + "\t");
			aux.adicionar(dat);
		}
		System.out.println();
		vaciar(aux);
	}
	void vaciar(ColaCircular z)
	{
		while (!z.esvacia())
			adicionar(z.eliminar());
	}
}
```

**ColeSImple**

```
import java.util.Scanner;

public class ColaSimple {
	private int max = 50;
	private int []V = new int [max + 1];
	private int ini, fin;

	ColaSimple()
	{
		ini = fin = 0;
	}
	boolean esvacia()
	{
		return ini == fin;
	}
	boolean esllena()
	{
		return fin == max;
	}
	void adicionar(int x)
	{
		if (!esllena())
		{
			V[fin + 1] = x;
			fin++;
		}
		else
			System.out.println("Esta Llena");
	}
	int eliminar()
	{
		int dat = -1;
		if (!esvacia())
		{
			ini++;
			dat = V[ini];
			if (fin == ini)
				ini = fin = 0;
		}
		else
			System.out.println("Cola Vacia");
		return dat;
	}
	void llenar(int n)
	{
		Scanner in = new Scanner(System.in);
		for (int i = 0; i < n; i++)
		{
			int x = in.nextInt();
			adicionar(x);
		}
	}
	void mostrar()
	{
		ColaSimple aux = new ColaSimple();
		System.out.println("Datos Cola: ");
		while (!esvacia())
		{
			int dat = eliminar();
			System.out.print(dat + "\t");
			aux.adicionar(dat);
		}
		System.out.println();
		vaciar(aux);
	}
	void vaciar(ColaSimple z)
	{
		while (!z.esvacia())
			adicionar(z.eliminar());
	}
	int nroElem()
	{
		return fin - ini;
	}
}
```

**PILA**

```
import java.util.Scanner;

public class Pila {
	private int max = 50;
	private int []V = new int [max + 1];
	private int tope;
	Pila()
	{
		tope = 0;
	}
	boolean esllena()
	{
		return tope == max;
	}
	void adicionar(int x)
	{
		if (!esllena())
		{
			V[tope + 1] = x; tope++;
		}
		else
			System.out.println("Pila Llena");
	}
	int eliminar()
	{
		int dat = -1;
		if (!esvacia())
		{
			dat = V[tope];
			tope--;
		}
		else
			System.out.println("Pila Vacia");
		return dat;
	}
	boolean esvacia()
	{
		return tope == 0;
	}
	void mostrar()
	{
		System.out.println("Datos Pila: ");
		Pila aux = new Pila();
		while (!esvacia())
		{
			int dat = eliminar();
			System.out.print(dat + "\t");
			aux.adicionar(dat);
		}
		System.out.println();
		vaciar(aux);
	}
	void llenar(int n)
	{
		Scanner in = new Scanner(System.in);
		for (int i = 0; i < n; i++)
		{
			int dat = in.nextInt();
			adicionar(dat);
		}
	}
	void vaciar(Pila z)
	{
		while (!z.esvacia())
			adicionar(z.eliminar());
	}
	int nroElem()
	{
		return tope;
	}
}
```

**MULTIPILA**

```
import java.util.Scanner;

public class MultiPila {
	private int np;
	private Pila v[] = new Pila[50];

	MultiPila()
	{
		for (int i = 1; i < 50; i++)
			v[i] = new Pila();
		np = 0;
	}
	boolean esvacia(int i)
	{
		return v[i].esvacia();
	}
	boolean esllena(int i)
	{
		return v[i].esllena();
	}
	int nroElem(int i)
	{
		return v[i].nroElem();
	}
	void adicionar(int i, int dat)
	{
		v[i].adicionar(dat);
	}
	int eliminar(int i)
	{
		return v[i].eliminar();
	}
	void llenar(int i, int n)
	{
		v[i].llenar(n);
	}
	void llenar(int n)
	{
		Scanner in = new Scanner(System.in);
		np = n;
		for (int i = 0; i < np; i++)
		{
			System.out.println("Intro nro de datos de la pila " + (i + 1));
			int num = in.nextInt();
			llenar(i + 1, num);
		}
	}
	void mostrar()
	{
		for (int i = 1; i <= np; i++)
			mostrar(i);

	}
	void mostrar(int i)
	{
		v[i].mostrar();
	}
	void vaciar(int i, Pila z)
	{
		v[i].vaciar(z);
	}
	void vaciar(int i, int j)
	{
		v[i].vaciar(v[j]);
	}
	public int getNp() {
		return np;
	}
	public void setNp(int np) {
		this.np = np;
	}

}
```

**MULTICOLA**

```
import java.util.Scanner;

public class MultiCola {
	int nc;
	private ColaSimple []v = new ColaSimple[50];

	MultiCola()
	{
		nc = 0;
		for (int i = 1; i < 50; i++)
			v[i] = new ColaSimple();
	}
	boolean esvacia(int i)
	{
		return v[i].esvacia();
	}
	boolean esllena(int i)
	{
		return v[i].esllena();
	}
	int nroElem(int i)
	{
		return v[i].nroElem();
	}
	void adicionar(int i, int dat)
	{
		v[i].adicionar(dat);
	}
	int eliminar(int i)
	{
		return v[i].eliminar();
	}
	void llenar(int i, int n)
	{
		v[i].llenar(n);
	}
	void llenar(int n)
	{
		nc = n;
		for (int i = 1; i <= nc; i++)
		{
			System.out.print("Intro nro de elementos de la cola" + i);
			Scanner in = new Scanner(System.in);
			int num = in.nextInt();
			llenar(i, num);
		}
	}
	void mostrar(int i)
	{
		v[i].mostrar();
	}
	void mostrar()
	{
		for (int i = 1; i <= nc; i++)
			mostrar(i);
	}
	void vaciar(int i, ColaSimple z)
	{
		v[i].vaciar(z);
	}
	void vaciar(int i, int j)
	{
		v[i].vaciar(v[j]);
	}
	public int getNc() {
		return nc;
	}
	public void setNc(int nc) {
		this.nc = nc;
	}

}
```

# LISTA CIRCULAR

```java
package ListaCircular;

public class Nodo {
	private Nodo ant, sig;
	private Numero num;

	Nodo()
	{
		ant = sig = null;
	}

	public Nodo getAnt() {
		return ant;
	}

	public void setAnt(Nodo ant) {
		this.ant = ant;
	}

	public Nodo getSig() {
		return sig;
	}

	public void setSig(Nodo sig) {
		this.sig = sig;
	}

	public Numero getNum() {
		return num;
	}

	public void setNum(Numero num) {
		this.num = num;
	}

}
```

```java
package ListaCircular;

public class ListaC {
	private Nodo p;

	ListaC()
	{
		p = null;
	}

	public Nodo getP() {
		return p;
	}

	public void setP(Nodo p) {
		this.p = p;
	}

	int nroNodos()
	{
		int cont = 0;
		Nodo w = getP();
		while (w != getP())
		{
			w = w.getSig();
			cont++;
		}
		cont++;
		return cont;
	}
	void mostrar()
	{
		Nodo w = getP();
		System.out.println("Datos Lista: ");
		while (w.getSig() != getP())
		{
			w.getNum().mostrar();
			w = w.getSig();
		}
		w.getNum().mostrar();
	}

}
```

```java
package ListaCircular;

public class ListaC {
	void adiFinal(Numero dat)
	{
		Nodo nuevo = new Nodo();
		nuevo.setNum(dat);
		if (getP() == null)
		{
			setP(nuevo);
			getP().setSig(nuevo);
			getP().setAnt(nuevo);
			getP().setSig(nuevo);
		}
		else
		{
			Nodo w = getP().getAnt();
			w.setSig(nuevo);
			nuevo.setAnt(w);
			nuevo.setSig(getP());
			getP().setAnt(nuevo);
		}
	}
	void leer1(int n)
	{
		for (int i = 0; i < n; i++)
		{
			Numero dat = new Numero();
			dat.leer();
			adiPrincipio(dat);
		}
	}
	Numero eliFinal()
	{
		Nodo r = getP().getAnt();
		if (nroNodos() == 1)
		{
			setP(null);
		}
		else
		{
			Nodo q = r.getAnt();
			q.setSig(getP());
			getP().setAnt(q);
		}
		r.setSig(null);
		r.setAnt(null);
		return r.getNum();
	}
}
```

# LISTA DOBLE

```java
package ListaDoble;

public class Nodo {
	Nodo ant, sig;
	Numero num;

	Nodo()
	{
		ant = sig = null;
	}

	public Nodo getAnt() {
		return ant;
	}

	public void setAnt(Nodo ant) {
		this.ant = ant;
	}

	public Nodo getSig() {
		return sig;
	}

	public void setSig(Nodo sig) {
		this.sig = sig;
	}

	public Numero getNum() {
		return num;
	}

	public void setNum(Numero num) {
		this.num = num;
	}

}
```

```java
package ListaDoble;

public class ListaD {
	private Nodo p;

	ListaD()
	{
		p = null;
	}

	public Nodo getP() {
		return p;
	}

	public void setP(Nodo p) {
		this.p = p;
	}

	int nroNodos()
	{
		Nodo w = getP();
		int cont = 0;
		while (w != null)
		{
			cont++;
			w = w.getSig();
		}
		return cont;
	}
	void adiprincipio(Numero dat)
	{
		Nodo nuevo = new Nodo();
		nuevo.setNum(dat);
		if (getP() == null)
			setP(nuevo);
		else
		{
			nuevo.setSig(getP());
			getP().setAnt(nuevo);
			setP(nuevo);
		}
	}
}
```

```java
package ListaDoble;

public class ListaD {
	void adifinal(Numero dat)
	{
		Nodo nuevo = new Nodo();
		nuevo.setNum(dat);
		if (getP() == null)
			setP(nuevo);
		else
		{
			Nodo w = getP();
			while (w.getSig() != null)
				w = w.getSig();
			w.setSig(nuevo);
			nuevo.setAnt(w);
		}
	}

	void leer1(int n)
	{
		for (int i = 0; i < n; i++)
		{
			Numero dat = new Numero();
			dat.leer();
			adiprincipio(dat);
		}
	}
	void leer2(int n)
	{
		for (int i = 0; i < n; i++)
		{
			Numero dat = new Numero();
			dat.leer();
			adifinal(dat);
		}
	}
	void mostrar()
	{
		Nodo w = new Nodo();
		System.out.println("Datos Lista: ");
		while (w != null)
		{
			w.getNum().mostrar();
			w = w.getSig();
		}
	}
	void elifinal()
	{
		Nodo w = getP();
		while (w.getSig() != null)
			w = w.getSig();

		Nodo q = w.getAnt();
		q.setSig(null);
		w.setAnt(null);
	}

}
```

**LISTA SIMPLE**

```java
package Listas;

public class Nodo {
	private Nodo sig;
	private Numero num;
	Nodo()
	{
		sig = null;
	}
	public Nodo getSig() {
		return sig;
	}
	public void setSig(Nodo sig) {
		this.sig = sig;
	}
	public Numero getNum() {
		return num;
	}
	public void setNum(Numero num) {
		this.num = num;
	}

}
```

```java
package Listas;

public class ListaS {
	Nodo p;
	ListaS()
	{
		p = null;
	}
	public Nodo getP() {
		return p;
	}
	public void setP(Nodo p) {
		this.p = p;
	}

	int nroNodos()
	{
		int c = 0;
		Nodo w = getP();
		while (w != null)
		{
			c++;
			w = w.getSig();
		}
		return c;
	}
	void mostrar()
	{
		Nodo w = getP();
		System.out.println("Datos Lista: ");
		while (w != null)
		{
			w.getNum().mostrar();
			w = w.getSig();
		}
	}
	void adifinal(Numero dat)
	{
		Nodo nuevo = new Nodo();
		nuevo.setNum(dat);
		if (getP() == null)
			setP(nuevo);
		else
		{
			Nodo w = getP();
			while (w.getSig() != null)
				w = w.getSig();
			w.setSig(nuevo);
		}
	}

}
```

```java
package Listas;

public class ListaS {

	void leer2(int n)
	{
		for (int i = 0; i < n; i++)
		{
			Numero num = new Numero();
			num.leer();
			adifinal(num);
		}
	}
	Numero eliFinal()
	{
		Nodo w = getP();

		while (w.getSig() != null)
			w = w.getSig();

		Nodo q = getP();
		while (q.getSig() != w)
			q = q.getSig();

		Numero dat = w.getNum();
		q.setSig(null);
		return dat;
	}
	Numero eliPrincipio()
	{
		Numero dat = getP().getNum();
		Nodo e = getP();
		setP(e.getSig());
		e.setSig(null);
		return dat;
	}
}
```