

P2 C3 2006

a)

```
Linea L;
String s,b;
for (int i=1;i<6;i++){
    if(i==3)i++;
    L=new Linea(i);
    U.println("Estaciones linea x");
    for(int k=1;true;k++){
        int c=L.combinacion(s);
        if(c!=0)b=" combina con línea
"+c;
        else c="";
        U.println(k+" "+s+b);
        s=L.siguiente(s);
    }
}
```

b)

```
String anterior(String s){
    if(primero.valor.equals(s))
        return null;
    // no tiene anterior
    Nodo x=this.primero;
    while(x.siguiente!=null&&
        !x.siguiente.valor.equals(s))
        x=x.siguiente;
    if(x.siguiente==null)return null;
    else return x.valor;
}
```

P3 C3 2006

a) //Vemos si existe alguna combinación entre sus padres, que sean hermanos

```
boolean primos(AG x, AG y){
    String xn=x.nombre(),
           yn=y.nombre();
    if(hnos(x.madre(xn),y.madre(yn),x,y)||hnos(x.madre(xn),y.padre(yn),x,y)
        hnos(x.padre(xn),y.madre(yn),x,y)||hnos(x.padre(xn),y.padre(yn),x,y)))
        return true;
    else return false;
}
boolean hnos(String nx, String ny, AG x, AG y){
    if(x.padre(nx).equals(y.padre(ny)) && x.madre(nx).equals(y.madre(ny)))
        return true;
    else
        return false;
}
```

b)

```
String madre(String x){
    String m=madrel(this.raiz,s);
    if(m.equals(""))return null;
}
String madrel(Nodo x, String s){
    if(x==null)return ""; // si la persona no existe
    if(x.nombre.equals(s))return x.madre.nombre;// si encuentro la persona
    return madrel(x.padre)+madrel(x.madre); //caso general
}
```

P3 C3 2004.

a)(ponderación: 1/3) Escriba un método de encabezamiento **void lugares(Torneo T)** que muestre la clasificación completa al final del torneo T de 64 jugadores en la forma: 1°X 2°X 3°X,X 5°X,X,X,X 9°X,X,X,X,X,X,X,X 17°... 33°... en que X es el nombre de un jugador

```
//metodo lugares: 2 ptos
static public void lugares(Torneo t){
U.println("1o"+t.jugador(1)); //0.5
for(int i=0; i<=5; ++i){ //0.5
    int n=(int)Math.pow(2,i)+1; //0.5
    U.println(n+"o"+t.jugador(n)); //0.5
}
}
```

b)(ponderación 2/3) Escriba el método **lugar** suponiendo que los resultados del torneo se registran en un árbol binario. Por ejemplo, el siguiente árbol binario muestra un torneo entre 4 en que el campeón fue B:

```
class Nodo{String valor; Nodo izq, der;}

public int lugar(String x){
    return lugar(x,1,raiz); //0.5
}
protected int lugar(String x,int n,Nodo r){//0.5
    if(r==null) return -1; //0.5
    if(r.valor.equals(x)) //0.5
        return n/2+1; //0.5
    return Math.max( //0.5
        lugar(x,2*n,r.izq), //0.5
        lugar(x,2*n,r.der)); //0.5
}
}
```

c)

```
static public void lugares(Torneo t){
Stack a=new Stack();
a.push("1o"+t.jugador(1));
for(int i=0; i<=5; ++i){
    int n=(int)Math.pow(2,i)+1;
    a.push((n+"o"+t.jugador(n)));
}
for(int i=0; i<=6; ++i){
    U.println(a.pop());
}
}

class Stack{
    Nodo primero;
    public Stack(){ primero=null; }
    void push(Object x){
        primero=new Nodo(x,primero);
    }
    Object pop(){
        Object x=primero.valor;
        primero=primero.sgte;
    }
}

class Nodo{
    Object valor;
    Nodo sgte;
    Nodo(Object x,Nodo s){
        valor=x;
        sgte=s;
    }
}
}
```