

## Listas circulares de enlace simple

```

public class Nodo {
    int info;
    Nodo enla;
    public Nodo(int x) {
        info = x;
        enla = null;
    }
}

public class ListC {
    Nodo cab;
    public ListC() {
        cab=null;
    }
    void Insertar(int x){
        Nodo aux=new Nodo(x);
        if(cab==null){
            cab=aux;
            cab.enla=cab;
        }
        else{
            Nodo tem=cab;
            while(tem.enla!=cab)
                tem=tem.enla;
            aux.enla=cab;
            tem.enla=aux;
        }
    }
    void Imprimir(){
        if(cab!=null){
            Nodo aux=cab;
            do{
                System.out.print(aux.info + " ");
                aux=aux.enla;
            }while(aux!=cab);
        }
    }
    /*esta es una forma muy complicada de imprimir datos en forma
    intercalada. Pero a la vez es muy interesante al utilizar
    aux.enla.enla*/
    void ImprimirIntercalado(){
        if(cab!=null){
            Nodo aux=cab;
            do{
                System.out.print(aux.info+" ");
                aux=aux.enla.enla;
            }while(aux!=cab&&aux.enla!=cab);
            if(aux!=cab)
                System.out.print(aux.info+" ");
            aux=cab.enla;
            if(cab.enla!=cab&&cab.enla.enla!=cab){
                do{
                    System.out.print(aux.info+" ");

```

```

        aux=aux.enla.enla;
    }while (aux!=cab&&aux.enla!=cab) ;
    }

    if (aux!=cab)
        System.out.print(aux.info+" ");
    }
}
}

```

```

import java.util.Scanner;
public class Prueba {

    public static void main(String[] args) {
        Scanner cs=new Scanner(System.in);
        int n=cs.nextInt();
        int x;
        ListC lc=new ListC();
        while(n!=0){
            n--;
            x=cs.nextInt();
            lc.Insertar(x);
        }
        lc.ImprimirIntercalado();
    }
}

```