

Solucion Examen Tipo B

Ejercicio 1

```
// devuelve un array formado por los valores mínimos de dos arrays pasados como parámetros
public static int[] menores(int[] a1, int[] a2) {
    int[] maximo, minimo, salida;
    // obtiene el array de mayor y menor longitud
    if(a1.length > a2.length){
        maximo = a1;
        minimo = a2;
    } else {
        maximo = a2;
        minimo = a1;
    }
    // crea un array de salida con el mismo tamaño que el mayor de los array de entrada
    salida = new int[maximo.length];
    // recorre el array de menor tamaño comparando los valores de los dos arrays
    for (int i = 0; i < minimo.length; i++){
        salida[i] = Integer.min(minimo[i], maximo[i]);
    }
    // una vez acabe el array de menor tamaño, los valores que quedan son
    // todos del array de mayor tamaño
    for (int i = minimo.length; i < maximo.length; i++){
        salida[i] = maximo[i];
    }
    // devuelve el array de salida
    return salida;
}
```

Ejercicio 2

```
// sobrecarga el método menores para poder utilizarlo con 3 arrays
public static int[] menores(int[] a1, int[] a2, int[] a3) {
    int[] salida;
    // compara los dos primeros y luego la salida con el tercero
    salida = menores(a1, a2);
    salida = menores(salida, a3);
    return salida;
}
```

Ejercicio 3

1)

```
public class Poligono {
    private String nombreForma;
    private int numLados;
    private int numVertices;
```

```
private double area;  
private boolean esCarpesiano;
```

```
// constructores
```

```
public Poligono(){  
    this.nombreForma = "";  
    this.numLados = 1;  
    this.numVertices = 1;  
    this.area = 1.0;  
}
```

```
Public Poligono(String nombreForma, int numLados, int numVertices, double area){  
    this.nombreForma = nombreForma;  
    this.numLados = numLados;  
    this.area = area;  
    this.numVertices = numVertices;  
    if (pi() > 25.0){  
        this.esCarpesiano = true;  
    } else {  
        this.esCarpesiano = false;  
    }  
}
```

```
// devuelve el valor de pi
```

```
private double pi(){  
    return (this.numLados * this.numVertices) / this.area;  
}  
}
```

2)

```
Poligono pl1 = new Poligono();  
Poligono pl2 = new Poligono("Cuadrado", 4, 4, 9);
```

Ejercicio 4

```
// algoritmo recursivo que muestra los números en orden creciente desde 1  
// hasta el valor pasado como parámetro  
public static void hastaN(int n){  
    if (n <= 1){  
        // si n es 1 deja de iterar.  
        System.out.print(n + " ");  
    } else { // itera con el número inferior y luego muestra  
        hastaN(n - 1);  
    }  
}
```

```
    System.out.print(n + " ");  
  }  
}
```