



Nº matrícula: _____ Nombre: _____

Apellidos: _____

Problema. Se pretende transportar una serie de productos ($p_1, p_2, p_3, \dots, p_N$) cada uno con un peso característico, desde la estación espacial conjunta hasta la base lunar internacional. Para ello se dispone de tres naves de carga a las que denominaremos n_1, n_2 y n_3 . Se dispone de la siguiente clase Java:

```
public class Producto {  
    private int nombre;  
    private int peso;  
  
    Producto (int nombre, int peso){  
        this.nombre = nombre;  
        this.peso = peso;  
    }  
    /* Conjunto de getters y setters */  
}
```

Se pide implementar un algoritmo en Java basado en un esquema Voraz que distribuya la carga entre las tres naves de la manera mas uniforme posible en lo relativo al peso. El método implementado deberá tener la siguiente cabecera:

```
void distribuyeCarga( ArrayList<Producto> productos, ArrayList<Producto> n1,  
                    ArrayList<Producto> n2, ArrayList<Producto> n3);
```

El parámetro *productos* representa un ArrayList que incluye los Objetos del tipo Producto a transportar a la Base Lunar y en los ArrayList *n1*, *n2* y *n3* se devolverá la relación de los productos a cargar en cada nave.

```
void distribuyeCarga(ArrayList<Producto> productos, ArrayList<Producto> n1,  
                    ArrayList<Producto> n2, ArrayList<Producto> n3){  
  
    int totalA=0, totalB=0, totalC=0;  
    Producto p;  
    while (!productos.isEmpty()){  
        p = seleccionarCandidato(productos);  
        productos.remove(p);  
        if (totalA <= totalB) // la nave n1 tiene menor carga que la nave n2  
        {  
            if (totalA<=totalC) { // n1 es la que menor carga tiene  
                n1.add(p);  
                totalA = totalA + p.getPeso();  
            }  
            else { // n3 es la que menor carga tiene  
                n3.add(p);  
                totalC = totalC + p.getPeso();  
            }  
        }  
    }  
}
```

```

    else if (totalB <= totalC) { // n2 es la que menor carga tiene
        n2.add(p);
        totalB = totalB + p.getPeso();
    }
    else { // n3 es la que menor carga tiene
        n3.add(p);
        totalC = totalC + p.getPeso();
    }
}
}

```

```

Producto seleccionarCandidato(ArrayList<Producto> candidatos){
    // selecciona el candidato que mayor valor tenga
    Producto p = candidatos.get(0);
    for (int i=1; i<candidatos.size();i++)
        if (p.getPeso() < candidatos.get(i).getPeso()) p=candidatos.get(i);
    return p;
}

```