2. Dado un conjunto X con $|X| = n$ y un entero $k \leq n$ queremos encontrar el máximo valor que pueden sumar los elementos de un subconjunto S de X de tamaño k. Más formalmente, queremos calcular $\max_{S \subseteq X, |S|=k} \sum_{s \in S} s$.

$X = \{x_1 ... x_n\}$, y tenemos un $k \leq n$

maximo subconjunto S de X de tamaño k

$X = \{1, 23, 3, 56, 2, 6\}$, $k = 4$

$S = \{56, 23, 6, 3\}$

a) Proponer un algoritmo *greedy* que resuelva el problema, demostrando su correctitud. Extender el algoritmo para que también devuelva uno de los subconjuntos S que maximiza la suma.

```
Algoritmo(X,k):
  X' <- sort(X) tal que ordena de mayor a menor

  res = 0

  para todo i: 1...k:
    res+= X'[i]

  return res
```

Invariante: la suma de los k elementos más grandes hasta la k-esima iteración

Tenemos un conjunto X' que es el conjunto X, ordenado de forma decreciente. $\forall i < j :: X'_i > X'_j$

Entonces los k primeros elementos son los mas grandes por el invariante. y la solución greedy G nos dice que: $G = \sum_{i=1}^{k} X'_i$

Tenemos una solucion optima O y sabemos que $O \leq G$ porque por invariante, G tiene los k elementos más grandes.

Pero O no puede ser peor que G porque O es optima, luego O no es menor a G, por lo que O es igual de optima que G