Exercici 7: Algorismes de dividir i vencer I.

Lliurament:

UN ÚNIC FITXER (exercici7.py) QUE CONTINGUI EL CONJUNT DE FUNCIONS QUE S'HAN IMPLEMENTAT.

- Escriu una funció no recursiva (**negatius**) tal que donada una llista de nombres enters (possiblement desordenats) la funció ha de retornar la mateixa llista amb els nombres negatius al capdavant i els positius al darrera (sense importar l'ordre entre ells). Exemple del comportament d'aquesta funció:

```
>>> a=[1,-2,3,-4,-3,5,6]
>>> negatius(a)
>>> a
[-6,-2,-3,-4,5,3,1]
```

Indicacions: Aquest algorisme es pot resoldre de la forma demanada amb una estratègia semblant (tot i que una mica més simple) a la *partició* del *quicksort*.

- Escriu una funció (**exponent**) que usant tècniques de dividir i vèncer, calculi el valor a^n per qualsevol $a>0$, sent n un enter positiu. Exemple del comportament d'aquesta funció:

```
>>> exponent(3,5) # a=3 , n=5
243
>>> exponent(3.5,5) # a=3.5 , n=5
525.21875
>>> exponent(45.78,45)
5.3753961414860256e+74
```

Indicacions: Podeu fer servir el fet que $a^n = a^{\lfloor n/2 \rfloor} \cdot a^{\lceil n/2 \rceil}$

- Escriu una funció (**reverse**) que usant tècniques de dividir i vèncer, inverteixi l'ordre dels caràcters d'un string. Exemple del comportament d'aquesta funció:

```
>>> reverse("hola, com estas?")
"?satse moc ,aloh"
```

- Donat un conjunt de n nombres reals [x1, x2, ..., xn] fes un algorisme, **aprop**, de dividir i conquerir que calculi quina és la parella de punts que estan més a prop un de l'altre. Calcula la seva complexitat.

- Donada una llista de n elements, escriu una funció, **elimina**, que elimini tots els elements duplicats amb complexitat O(nlog n).