Auxiliar

Problema 1

En esta pregunta queremos ordenar números distintos entre 0 y 32767. Dado que sabemos que existen métodos eficientes de ordenamiento por comparación que toman tiempo $O(n \log n)$, le pedimos a usted que ordene (como se le ocurra hacerlo) los números entre 0 y 32767 en tiempo $O(n)$. Suponga que el final del arreglo se indica con un elemento igual a -1 (que no se ordena, pues sólo indica que el arreglo terminó).

Para hacer el problema más *entretenido* sólo se le permite usar 4 kilobytes de memoria y para ser generosos un máximo de 5 variables locales. No puede usar los siguientes operadores: *,$,/, >, <, == para realizar su función de ordenamiento.

```
void bucketsort(int *a);
```

Problema 2

Para verificar errores de transmisión, se usa un bit de paridad que indica si el número de bits en uno de una secuencia es par o impar. Se les pide implementar dos funciones quepermitan manejar bits de paridad en un byte, suponiendo que los 7 bits de orden inferior son datos y el bit 8 de orden superior es un bit de paridad. La regla es: si el número debits en uno de los 7 inferiores es par, el octavo bit debe ir en 1. Si es impar debe ir en 0.

```
unsigned charset_parity(unsigned charc);
int check_parity(unsigned char c);
```