

```
def min_heapify(array, i):
    #min_heapify aldığı parametler-> array:heap dizisi, i: i'inci eleman
    #bu fonksiyon dizinin i'inci elemanı ile o elemanın child larını
    karşılaştırarak küçük olanı yukarı taşıyıp doğru heap dizimini sağlar.
    left= 2*i + 1
    right= 2*i + 2
    length= len(array)-1
    smallest = i
    if left <= length and array[i] > array[left]:
        smallest = left
    if right <= length and array[smallest] > array[right]:
        smallest = right
    if smallest != i:
        array[i], array[smallest] = array[smallest], array[i]
        min_heapify(array, smallest)

def build_min_heap(array):
    #aldığı parametre-> array: dizi
    #bu fonksiyon tüm elemanlara min_heapify uygular ve verilen diziyi heap
    haline getirir.
    for i in reversed(range(len(array)//2)):
        min_heapify(array, i)

def heapsort(array):
    #aldığı parametre-> array:heap dizisi | gönderdiği parametre->
    sorted_array:sıralanmış dizi
    #bu fonksiyon verilen heap dizisini sıralı biçimde geri döndürür.
    array = array.copy()
    build_min_heap(array)
    sorted_array = []
    for _ in range(len(array)):
        array[0], array[-1] = array[-1], array[0]
        sorted_array.append(array.pop())
        min_heapify(array, 0)
    return sorted_array

myArray = [8,10,3,4,7,15,1,2,16]
build_min_heap(myArray)
print(myArray)
```

```
def insertItemToHeap(myheap_1,item):
    #aldığı parametreler-> myheap_1:heap dizisi, item:heap dizisine eklenecek
    değer.
    #bu fonksiyon parametreden aldığı sayıyı yine aldığı heap dizisine ekler
    sonra da build_min_heap fonksiyonu yardımıyla yeniden heap haline getirir.
    if(myheap_1.count(item)==0):
        myheap_1.append(item)
        build_min_heap(myheap_1)

insertItemToHeap(myArray,42)
print(myArray)
```

```
def removeItemFrom(myheap_2):
    #aldığı parametreler-> myheap_1:heap dizisi | gönderdiği parametreler
    myheap_2:elemanı silinmiş yeni heap dizisi
    #bu fonksiyon dizinin uzunluğu 0 değil ise root(kök) elemanı ile son
    elemanın yerlerini değiştirip yeni son elemanı siler ve build_min_heap ile
    tekrar heap haline getirir.
    if(len(myheap_2)):
        #myheap_2 = heapsort(myheap_1)
        myheap_2[0], myheap_2[-1] = myheap_2[-1], myheap_2[0]
        myheap_2.pop()
        build_min_heap(myheap_2)
        return myheap_2

removeItemFrom(myArray)
removeItemFrom(myArray)
```