Q1] Consider the following function that takes a list of ints. What does the function do and what is the time complexity of it? The function randomFunc1 has 1 parameter x that is a list of ints.

```
def randomFunc(x):
        a = []
        for i in range(len(x)):
        b = 1
        for j in range(len(x)):
            if j != i and (j % 2 == 0):
                    b = b * x[j]
        a.append(b)
    return a
```

Q2] Consider the following function that takes a list of ints. What does the function do? Rewrite the function recursively.

The function randomFunc2 has 1 parameter x that is a node of a tree. Assume each node has attributes left, right and data.

```
def randomFunc2(x):

    if x is None:
        return
    y = []
    z = []
    y.append(x)
    while (len(y) >0):
        k = y.pop()
        z.append(k)
        if k.left is not None:
            y.append(k.left)
        if k.right is not None:
            y.append(k.right)
    while(len(z) > 0):
        k = z.pop()
        print k.data
```

Q3] Code heapify for a priority heap and find the complexity.
Hint: What is the height of a compact tree?
Hint: The index of the left child of the node at index i is 2i+1.
Hint: The index of the right child of the node at index i is 2i+2.
Hint: The index of the parent of the node at index i is (i-1)//2. [Integer division]