Lab Exercise 2

Implement following functions using recursion.

1) flip :: [(a,b)] ->[(b,a)] which flips each pair in the list.

```
>>flip [(1,'a'),(2,'b')]
 [('a',1), ('b',2)]
```

2) dupli :: [a] -> [a] which duplicate each elements of a list.

```
>>dupli [1,2,3]
[1,1,2,2,3,3]

>>dupli ['a','b','b']
['a','a','b','b','b','b']
```

3) vowels :: [Char] -> [Char] that takes a String and returns all the vowels in the string

```
>>vowels "hello"
"eo"
```

4) repli :: [a] -> Int -> [a] which replicate the elements of a list a given number of times.

```
>>repli [1,2,3] 3
[1,1,1,2,2,2,3,3,3]

>>repli "abc" 4
"aaaabbbbcccc"
```

5) doubleEven :: [Int] -> [Int] which doubles each even numbers.

```
>>doubleEven [2,3,5,6]
[4,3,5,12]
```

6) elem' :: [a] -> a -> Bool which takes a list xs and an element x, and returns True if x is an element of xs.

```
>>elem' [1,2,3] 2
True

>>elem' "abcd" 'e'
False
```

7) kthElem :: [a] -> Int -> a that takes a list and an integer k and returns the kth element of the list, where elements are numbered starting from 0. (Do not use the built in operator !!)

```
>>kthElem ['h','e','l','l','o'] 1
'e'
```

8) riffle :: [a] -> [a] -> [a] that takes two lists of same length, and interleaves their elements in turn about order.

```
>>riffle [1,2,3] [4,5,6]
[1,4,2,5,3,6]
```

9) rotate :: [a] -> Int -> [a] that takes a list and an integer n and rotates the list n places to the left.

```
>>rotate ['a','b','c','d','e'] 2
```

['c','d','e','a',b']

10) merge :: Ord a => [a] -> [a] -> [a] that merges two sorted lists to give a single sorted list.

```
>>merge [2,5,6] [1,3,4]
[1,2,3,4,5,6]
```

11) mergesort :: Ord a => [a] -> [a] that sorts the given list.(Use merge)

12) dropEvery :: [a] -> Int -> [a] that takes a list and an integer n and drops every nh element from the list.

```
>>dropEvery [1,2,3,4,5,6,7,8,9,10] 3
[1,2,4,5,7,8,10]
```