Joanna Li

Hw3

Part 1: pseudo code for a few recursive functions, continuing what is discussed in class on Monday.

addAllRec

```
Input: int[] arr, int n
Output: int sum
addAllRec( int[] arr, int n)
    if n == 1
        return a[0]
    else
        return addAllRec(arr, n-1) + arr[n-1]
```

productAllRec

```
Input: int[] arr, int n
Output: int product
productAllRec(int[] arr, int n)
    if n == 1
        return a[0]
    else
        return productAllRec(arr, n-1) * arr[n-1]
```

isElementRec

```
Input: int[] arr, int n
Output: boolean isElement
isElementRec(int[] arr, int n, int i, int value)
    if i == n
        return false
    if arr[i] == value
        return true
    else
        return isElementRec(arr, n, i+1, value)
```

findMinRec

```
Input: int[] arr, int n
Output: int min
findMinRec(int[] arr, int n)
    if n == 1
        return arr[0]
    int min = findMinRec(arr, n - 1)
    int currentInt = arr[n - 1]

    if currentInt < min
```

```
        return currentInt
    else
        return min
```

findMaxRec

Input: int[] arr, int n

Output: int max

```
findMaxRec(int[] arr, int n)
    if n == 1
        return arr[0]
    int max = findMaxRec(arr, n - 1)
    int currentInt = arr[n - 1]

    if currentInt > max
        return currentInt
    else
        return max
```

reverseRec

Input: int[] arr, int n

Output: int[] reversedArr

```
reverseRec(int[] arr, int i , int n )
    if i > n
        return arr
    int tempInt = arr[i]
    arr[i] = arr[n-1]
    arr[n-1] = tempInt
    return reverseRec(i + 1, n - 1)
```