Jarlin Almanzar 12/9/16
CS 111 Assignment 5

Problem 1:

1. Selection-sort Unsorted Array(Into alphabetical order):
Initial array: X A T B Q S B

1. X A T B Q S B Swap X and A
2. A X T B Q S B Swap X and B
3. A B T X Q S B Swap T and B
4. A B B X Q S T Swap X and Q
5. A B B Q X S T Swap X and S
6. A B B Q S X T Swap X and T
7. A B B Q S T X //Sorting has been completed

Sorted Array: A B B Q S T X
The amount of comparisons made is: 6 swaps

2. Insertion-sort Unsorted Array:

X A T B Q S B
Place X in 1st place
1: X - A T B Q S B
A < X, insert A before X
2: A X - T B Q S B
T > A, and smaller than T, so insert T before X and after A
3: A T X - B Q S B
B > A, and smaller than T, so insert B before T and after A
4: A B T X - Q S B
B = B, so insert B after B
5: A B B T X - Q S
Q < T and bigger than B, so insert Q before T and after B
6: A B B Q T X - S
S > Q and smaller than T, so insert S before T and after Q
7: A B B Q S T X -
X is bigger than all the other elements, leave it as it is

Sorted Array: A B B Q S T X
The amount of comparisons is: 7 swaps

Problem 2:

a. Determine if 2 arrays contain the none of the same elements (assume all elements are distinct)
1. Choose the 1st element of the 1stArray, and then search for it in the 2nd array
The increment the 1st element to the 2nd after first check(if everything was unique) and so on.

```
For the code we will need two for loops:
        The outer for loop is for the 1st array and 2nd array is the inner loop:
For(int n1=0; i<arr1.length; i++)
    For(int n2=0; j<arr2.length; j++)
        If(arr1[n1]==arr2[n2])
                Its same;
                Return false;
        Else
                Not same;
                Return true; //distinct
```

Both loops go through n iteration.
The number of iterations is: n*n+1 or O(n^2)
The best case Input: If first comparison both elements are the same as it only compares 1 time, and returns false immediately
The worst case Input: You go to through both loops and same elements is found at the last

iteration or you go through both loops and find that all elements are distinct

b. Counting total number characters that have a duplicate within a string (i.e. "gigi the gato" would result in 7 (g x 3 + i x 2 + t x 2)

1. Choose the first letter in the string

```
We will need two for loops
For(int n1=0; n1<str1.length(); n1++)
        For(int n2=1; n2<str2.length(); n2++)
                If(str1.charAt(n1)==str2.charAt(n2))
                        Same;
                        Count++;
                newStr +=count;
                Count=0;
```

Both loops go through iterations

The number of iterations is: n^3 since we have to compare n times with n^2 in the substring check O(n^3) or it can be just O(n) if

The Best case input: O(n^3) when everything has to be checked and go into the if comparison

The Worst case input: O(n) if nothing goes in the substring check

c. Finding a row where every entry is 'a' in a 2-D array.

For a 2d array we have Rows and Cols

```
For(rows)
        For(cols)
                If([rols][cols]==’a’)
                Count++;
                        If(count= rows.length)
                                Return row;
Count=0;
```

The double array will rely on N*M since it will have to check probably every row

Best Case input: O(n*m) if the first row contains all the ‘a’

Worst Case input: O(n*M) if the last row contains all the ‘a’ or there’s no row full of□a’