Adi Lind 322493107

ArrayOps-

```
public class ArrayOps {
    public static void main(String[] args) {
        int[] intargs = convertStringToIntArray(args);
        //int x = secondMaxValue(intargs);

        int[] a  = {0};
        int[] b = {0,1,1,1,0,3,0,1};
        System.out.println(findMissingInt(a));
    }

    public static int findMissingInt (int [] array) {
        // Write your code here:
        int length = array.length;
        int testnum = 0;
        if (length == 1)
        {
            if(array[0] == 1)
            {
                return 0;
            }
            return 1;
        }
        for(int i=0; i<length;i++)
        {
            Boolean flag = false;
            testnum = i;
            for(int j = 0; j< array.length ; j++)
            {
                if(array[j] == testnum)
                {
                    flag = true;
                    break;
                }
            }
            if (!flag)
            {
                break;
            }
        }
```

```
        return testnum;
    }

    public static int secondMaxValue(int [] array) {
        // Write your code here:
        int max = Math.max(array[0], array[1]); // get the max value for the 2
first objects
        int secmax = Math.min(array[0], array[1]); // get the min value
        for ( int i=2; i< array.length ; i++)
        {
            if(array[i] >= max)
            {
                secmax = max;
                max = array[i];
            }
            else if ( (array[i] > secmax) && (array[i] <= max))
            {
                secmax = array[i];
            }

        }
        return secmax;
    }

    public static boolean containsTheSameElements(int [] array1,int [] array2)
{
        // Write your code here:
        //int length1 = array1.length;
        //int length2 = array2.length;
        //Boolean flag1 = false , flag2 = false;
           // flag1 = contains(array1, array2);
            //flag2 = contains(array2, array1);


        //return (flag1&&flag2);
        for (int i = 0; i<array2.length ; i++) //test if array2 is in array1
        {
            if(!contains(array1, array2[i]))
            {
                return false;
            }
        }
        for (int k = 0; k<array1.length; k++) //test if array1 is contain
array2
        {
            if(!contains(array2, array1[k]))
            {
                return false;
```

```
            }
        }

        return true;
    }

    public static boolean isSorted(int [] array) {
        // Write your code here:
        if(array[0] <= array[1])
        {
            return checkIfIncreasing(array);
        }
        return checkIfDecreasing(array);
    }

    public static int[] convertStringToIntArray (String[] stringArray)
    {
        int length = stringArray.length;
        int[] intArray = new int[length];

        for ( int i = 0; i < length; i++)
        {
            intArray[i] = Integer.parseInt(stringArray[i]);
        }

        return intArray;
    }

    public static boolean contains(int[] array, int x)
    {
        for(int i = 0; i< array.length; i++)
        {
            if(array[i] == x)
            {
                return true;
            }
        }
        return false;

       }

       public static boolean contains (int[] array1, int[] array2) {
        Boolean flag = false;

        for(int k =0; k < array1.length ; k++)
            {
                int temp = array1[k];
                if(contains(array2, temp))
```

```
                {
                    flag = true;
                }
            }
            return flag;
        }

        public static boolean checkIfIncreasing (int[] array)
        {
         int temp = array[0];
         for(int i=1; i<array.length; i++)
         {
             if(array[i] < temp)
             {
                 return false;
             }
             temp = array[i];

         }
         return true;
        }

        public static boolean checkIfDecreasing (int[] array)
        {
         int temp = array[0];
         for(int i=1; i < array.length; i++)
         {
             if(array[i] > temp)
             {
                 return false;
             }
             temp = array[i];

         }
         return true;
        }
}
```

StringOps-

```
public class StringOps {
    ////////////////////////////////////////////////////////
    //////                                            ///////
    //////              Reminder:                     ///////
    //////        allowed methods                     ///////
    //////                                            ///////
    //////        1.charAt(int index)                 ///////
    //////        2.length()                          ///////
    //////        3.substring(int start)              ///////
    //////        4.substring(int start,int ends)     ///////
    //////        5.indexOf(String str)               ///////
    //////                                            ///////
    //////        The rest are not allowed !          ///////
    //////        if you want to use a different      ///////
    //////        method, and you can implement       ///////
    //////        it using material from the course   ///////
    //////        you need to implement a version of  ///////
    //////        the function by yourself.           ///////
    //////                                            ///////
    //////        see example for substring           ///////
    //////        in Recitation 3 question 5          ///////
    //////                                            ///////
    ////////////////////////////////////////////////////////
    public static void main(String[] args) {
        String test = "Hello world";
        System.out.println(allIndexOf(test,'l'));

    }

    public static String capVowelsLowRest (String string) {
        // Write your code here:
        String result = ""; // create the string we want to get
        String semiresult = lowerCase(string); // change the original string
we got to without Upper-Case
        int length = string.length();
        for(int i = 0 ; i< length ; i++)
        {
            char temp = semiresult.charAt(i);
            switch (temp) {
                case 97: //a
                    temp = (char) (temp - 32); //change a to A
                    result = result + temp;
                    break;

                default:
                    result = result + temp;
```

```
                    break;
                case 101: //e
                temp = (char) (temp - 32); //change e to E
                    result = result + temp;
                    break;

                case 105: //i
                temp = (char) (temp - 32); //change i to I
                    result = result + temp;
                    break;
                case 111: //o
                temp = (char) (temp - 32); //change o to O
                    result = result + temp;
                    break;

                case 117: //u
                temp = (char) (temp - 32); //change u to U
                    result = result + temp;
                    break;
            }
        }

        return result;
    }

    public static String camelCase (String string) {
        // Write your code here:
        int pointer = 0;
        while(string.charAt(pointer) == ' ') //count how many space we have
before the first word
        {
            pointer++;
        }
        //int pointer = string.indexOf(" ");
       // String semiresult = ""; //lowerCase(string.substring(0, pointer));
//help us to make the first word without uppercase
        String result = ""; // the string we will return
        int length = string.length();
        //semiresult = lowerCase(semiresult);
        //result += string.charAt(pointer); //get the first word without
uppercase
        //pointer++;
        if (pointer >= 1){ //edge problem that fix the problem of the first
word (needs to be only lowercases)
            if(string.charAt(pointer-1) == ' ' )
            {
                if(((int) string.charAt(pointer) <= 122) && ((int)
string.charAt(pointer) >= 97)) //if lowercase than put in the string
```

```
                {
                    result += result + string.charAt(pointer);
                }
                if(string.charAt(pointer)>= 65 && string.charAt(pointer) <=
90) // the ascii code of upper
                {
                    char temp2 = (char) (string.charAt(pointer) + 32);
//change to lowercase
                    result = result + temp2;
                }
            }
        }
        //after checking the spaces and the first char of the first word now
we check all the string
        while (length - pointer > 0)
        {
            if(string.charAt(pointer) == ' ')
            {
                while(string.charAt(pointer+1) == ' ')
                {
                    pointer++;
                }
                if(((int) string.charAt(pointer+1) <= 122) && ((int)
string.charAt(pointer+1) >= 97))
                {
                    char temp = (char)(string.charAt(pointer+1) - 32);
//change the first char after space to uppercase
                    result = result + temp;
                    pointer= pointer +1;
                }
                else //if the char after space is Uppercase so stay the same
                {
                    result = result + string.charAt(pointer+1);
                }
               if(string.charAt(pointer) == ' ')
                {
                    pointer= pointer +2;
                }

            }
            if (string.charAt(pointer)>= 65 && string.charAt(pointer) <= 90)
// the ascii code of upper
            {
                char temp2 = (char) (string.charAt(pointer) + 32); //change to
lowercase
                result = result + temp2;
            }
            else //is already lowercase so he is good
```

```
            if(string.charAt(pointer-1) != ' ')
            {
                result = result + string.charAt(pointer);
            }
            pointer++;
        }
        return result;
    }

    public static int[] allIndexOf (String string, char chr) {
        // Write your code here:
        String str = string;
        String newstr = "";
        //int[] arr = new int[str.length()];
        for(int i =0 ; i < str.length() ; i++)
        {
            if(str.charAt(i) == chr)
            {
                newstr += i;
            }
        }
        int[] arr = new int[newstr.length()];
        for(int j =0 ; j< newstr.length() ; j++)
        {
            int temp = newstr.charAt(j) -48; //take the value of the number in
ascii and change to regular int
            arr[j] =(temp);
        }
        return arr;
    }

    public static String lowerCase(String s) { //from the HW3
        int size = s.length();
        String newS = ""; //make the new s we get but without Upper-case.
        for(int i = 0 ; i < size; i++)
        {
            char temp = s.charAt(i);
            if (temp >= 65 && temp <= 90) // the ascii code of upper
            {
                temp = (char) (temp + 32) ;
                newS = newS + temp;
            }
            else
            {
                newS = newS + temp;
            }

        }
```

```
        return newS;
    }

}
```