Interpretation Worksheet

**Problem 1.0.** What does the following code print when the following input is used?

Hi There Friends!

```java
public class interpret01 {
    private String [] input;

    public Interpret01 (Scanner sc){
        input = new String [sc.nextInt()];
        for (int i = 0; i < input.length; i++)
            input[i] = sc.next();
        sc.close();
    }

    public char charAt (int aIndex, int sIndex){
        return input[aIndex].charAt(sIndex);
    }

    public int [] indexOf (char c){
        int [] indices = new int[2];
        for (int a = 0; a < input.length; a++)
            for (int s = 0; s < input[a].length(); s++)
                if (input[a].charAt(s) == c){
                    indices[0] = a;
                    indices[1] = s;
                }
        else{
            indices[0] = -1;
            indices[1] = -1
        }
        return indices;
    }

    public static void main (String [] args){
        Scanner scanner = new Scanner (System.in);
        Interpret01 a = new Interpret01(scanner);
        System.out.println(a.charAt(1, 1));
        System.out.println((Arrays.toString(a.indexOf('z'))));
        System.out.println((Arrays.toString(a.indexOf('e'))));
    }
}
```

**Problem 1.1.** What does the following code print?

**Problem 1.2.** In one sentence explain what the charAt method does:

**Problem 1.3.** In one sentence explain what exactly the indexOf method does (be careful, this is an incorrect implementation - what makes it incorrect?):
Problem 2.0. Here is a different implementation of the index of method:

```java
public int [] indexOf (char c){
    int [] indices = {-1, -1};
    for (int a = 0; a < input.length; a++)
        for (int s = 0; s < input[a].length(); s++)
            if (input[a].charAt(s) == c){
                indices[0] = a;
                indices[1] = s;
                return indices;
            }
    return indices;
}
```

Problem 2.1. Describe what this implementation does:

Problem 2.2. How does this method change when I delete line number 8?

Problem 3.0. public class ReferenceType {
    private int a = 10;
    private int [] b = {10, 20, 30, 40};

    public int getA(){
        return a;
    }

    public int [] getB(){
        return b;
    }

    public void add10(int c, int [] d){
        a += 10;
        c += 10;
        for (int i = 0; i < b.length; i++){
            b[i] += 10;
        }
        for (int i = 0; i < d.length; i++){
            d[i] += 10;
        }
    }

    public static void main(String [] args){
        int myLocalInt = 20;
        int [] myLocalArray = {20, 30, 40, 50};
        ReferenceType ref = new ReferenceType();
        ref.add10(myLocalInt, myLocalArray);
        System.out.println(myLocalInt);
        System.out.println(Arrays.toString(myLocalArray));
        System.out.println(ref.getA());
        System.out.println(ref.getB());
    }
}
Problem 3.1. What is printed?

Problem 3.2. Why do I use an object of type ReferenceType to print the methods in the main class?

Problem 3.3. Explain why no parameters are used when ref is created:

Problem 4.0. What does the following code print?

```java
public class ArraySort {

    public static void main(String[] args) {
        int[] a = new int[]{45, 12, 78, 34, 89, 21};
        Arrays.sort(a);
        for (int element : a){
            System.out.println(element);
        }
    }
}
```