Trace Code

For each problem below, answer the question(s) about the code.

a. What is the value of the array (what is stored in its memory locations) after these lines execute?

```java
int [] x = { 7, 9, -2, 4 };
x[2] = -x[2];
```

b. What are the values of each variable (what is stored in their memory locations) after these lines execute?

```java
double [][] dArray = new double[2][3];
dArray[0][2] = 4.5;
dArray[1][1] = 3.5;
dArray[1][2] = dArray[0][2] - dArray[1][1];
```

c. What are the values of each variable (what is stored in their memory locations) after these lines execute?

```java
String s = "hello";
char c = s.charAt(2);
c++;
s = "he" + c + c + "o";
```
d. What are the values of each variable (what is stored in their memory locations) after these lines execute?

```java
String t = "hee dee hee dee hee dee hee";
int x = t.indexOf("dee");
String u = t.substring(0, x);
u = u + " haw";
```

e. What are the values of each variable (what is stored in their memory locations) after these lines execute?

```java
String s = "hi dee hi dee hi dee hi";
String t = "";
int x = s.indexOf("hi dee", 0);
while(x>=0) {
    t = t + "ho dee";
    x = s.indexOf("hi dee", x+1);
}
t = t + "ho";
```

f. What are the values of each variable (what is stored in their memory locations) after these lines execute?

```java
String s = "hibbity hibbity" + " hibbity hibbity";
int i=0;
int count = 0;
while(i<s.length()-3) {
    if(s.substring(i, i+3).equals("bit")) {
        count++;
    }
    i++;
}
```
2. **2D Array programming**
   a. Declare a 2D Array of ints with some non-zero integers stored inside. Write a loop to compute the sum of the integers stored in the array. Print the average of the numbers on the screen.

   b. Declare a 2D array of chars with some non-zero chars stored inside. Write a loop to display the contents of the array on the screen, one row of the array per row of the screen.

3. **String programming**
   a. Declare and initialize some String variable \( s \) (any String value will do). Write a loop to display each character of the String on a separate line of the screen.

   b. Declare and initialize some String variable \( s \) (any String value will do). Declare another String variable called \( \text{reverse} \), initialized to the empty String “”. Write a loop so that by the end of the loop, \( \text{reverse} \) contains the reverse of the String value stored in \( s \).

   c. Declare and initialize some String variable \( s \) (any String value will do). Declare an int variable \( \text{indexOfHello} \). Write some code so that at the end of the code, \( \text{indexOfHello} \) will contain the index of the first occurrence of “hello” in \( s \), or -1 if “hello” is not contained in \( s \).