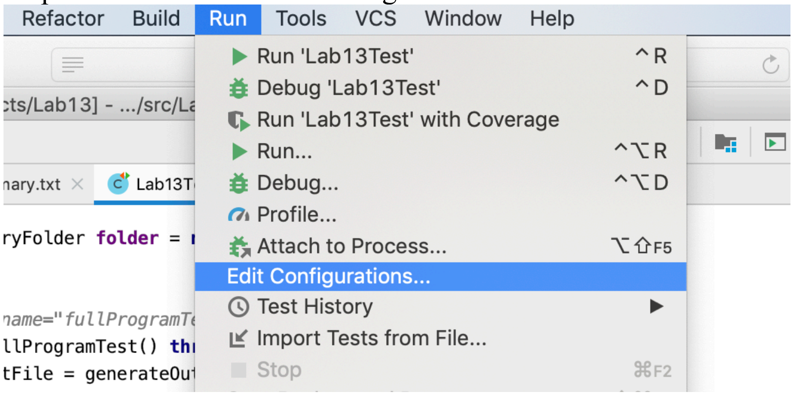
Lab 8 - Arrays

1. In IntelliJ, navigate to the **Labs** package and create another package named **Lab8**

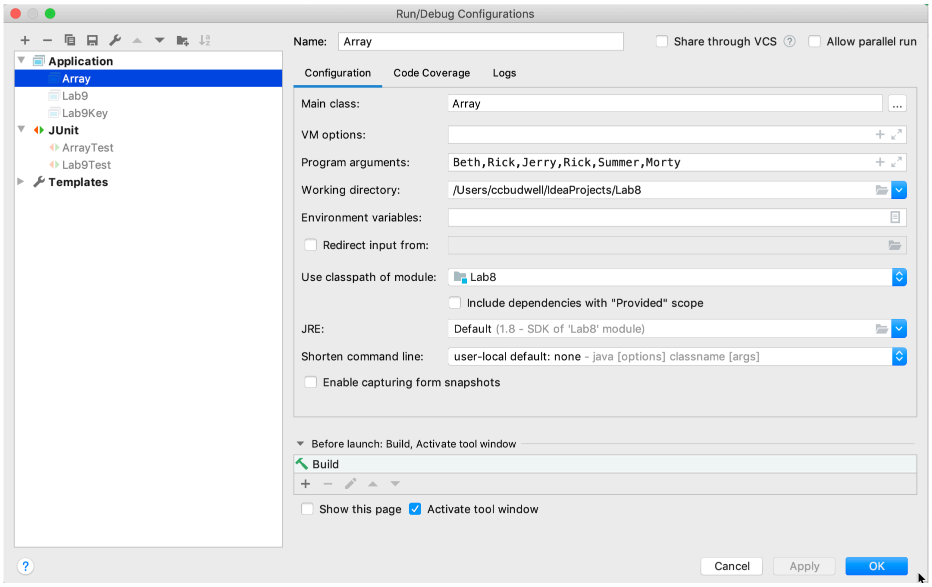
2. Create a class in the **Lab8** package called `Array.java` with a `main` method that will do the following:

   a. Input at the **<u>command line</u>** the following String: “Beth,Rick,Jerry,Rick,Summer,Morty” (DON”T type the “ ”.)

      i. Step 1: Go to Run → Edit Configurations



      ii. Step 2: Go to Application, you might need to use the + key to find it, and add the program arguments needed.



Name: Array □ Share through VCS □ Allow parallel run

Main class: Array

VM options:

Program arguments: Beth,Rick,Jerry,Rick,Summer,Morty

Working directory: /Users/ccbudwell/IdeaProjects/Lab8

Environment variables:

□ Redirect input from:

Use classpath of module: Lab8

□ Include dependencies with "Provided" scope

JRE: Default (1.8 - SDK of 'Lab8' module)

Shorten command line: user-local default: none - java [options] classname [args]

□ Enable capturing form snapshots

Before launch: Build, Activate tool window

Build

□ Show this page ☑ Activate tool window

   b. Use this inputted String and the method `.split(",")` to create an array of `String` objects called `tokens`. The actual code you will use is below:

```
String[] tokens = args[0].split(",");
```

   c. Create another String array called `customerName`. This array will have eight `String` objects.

   d. Create a for loop to copy each element in the `tokens` array into the `customerName` array. At this point the last two positions in the tokens array will be null.

0	Beth
1	Rick
2	Jerry
3	Rick
4	Summer
5	Morty
6	
7	

e. Write an `foreach` loop (aka an enhanced `for` loop) to display the array of names so that each name is on a separate line. End your output with a blank line. Answer the following questions:
   - What is displayed for the last two array elements?
   - Why is it that value?

f. Move the existing elements from indexes 4 and 5 to indices 6 and 7 respectively. You can then add the Strings “Rick” and “Jessica” into index 4, and 5, respectively. The array should then contain the following elements in this order:

0	Beth
1	Rick
2	Jerry
3	Rick
4	
5	
6	Summer
7	Morty

→

0	Beth
1	Rick
2	Jerry
3	Rick
4	Rick
5	Jessica
6	Summer
7	Morty

g. Write another `foreach` loop (aka an enhanced `for` loop) to display the array of names. Again, end your output with a blank line.

h. Reverse the elements in the array without creating a new array. Your new array should look like the array below. Hint: Think about this switching. How many switches do you need to make? Do you need to walk through the entire array?

0	Morty
1	Summer
2	Jessica
3	Rick
4	Rick
5	Jerry
6	Rick
7	Beth

i. Write another `foreach` loop (aka an enhanced `for` loop) to display the array of names. Again, end your output with a blank line.

j. Write a second, traditional `for` loop that checks each element for the String “Rick”, if found in the array, remove it, shift the remaining elements up, leaving null at the last position.

   Note: you will fill in the last element of the array to **null**, with no quotes, after shifting the elements up. You will also have to check that the **customerName[i] != null** before seeing if the value of customerName[i].equals(“Rick”).

k. Write another `foreach` loop (aka an enhanced `for` loop) to display the array of names. Again, end your output with a blank line.

l. Answer these questions:
   - Are both instances of “Rick” removed correctly from the array?

     Why or why not? Can you explain what you observe in your code? Why do you still have one Rick?

3. Run the JUnit test cases to make sure each one passes.

4. Take a screenshot of the `ArrayTest.java` running showing that all test cases pass.

5. Upload the `Array.java` file to Gradescope and ensure that all tests pass.

6. Upload the `Array.java` file and the screenshot to the submission area in Canvas.