You have seen how primitive values such as integers, floats, and characters are represented in memory at a machine level. You will also be studying how more complex data structures are represented in memory and accessed at the machine level. One of the simplest of these is an array.

The following code in Java defines an array of strings of equal length, and uses a main program and a procedure to access the array.

Comment the Java program, and then write the equivalent program in MIPS.

NOTES:
In Java, it is not necessary to pre-define the length of an array or the length of the strings in the array (as shown below), because length is an instance variable for an array in Java, and a method for the String class. However, the underlying machine code must somehow store those values with the array (so we have included the explicit variables here to make it easier to translate to MIPS).

It is also not necessary to have a compareStrings method/procedure, because the Java equals method for the String class can perform this operation. This program shows how the equals method might be implemented (again, to make it easier to translate to MIPS).

```java
public static int length_validops = 5;
public static int length_op = 3;
public static String[] validops = {"add","sub","and","slt","beq"};

public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter a string: ");
    String s = scan.next(); //read in a string from the keyboard

    for (int i = 0; i < length_validops; i++) {
        if (compareStrings(s,validops[i]) != 0) {
            System.out.println("Valid operation");
            break;
        }
    }
}

public static int compareStrings(String s1,String s2) {
    if (s1.length() != length_op) {
        return 0;
    }
    for (int i = 0; i< length_op; i++) {
        if (s1.charAt(i) != s2.charAt(i)) {
            return 0;
        }
    }
    return 1;
}
```