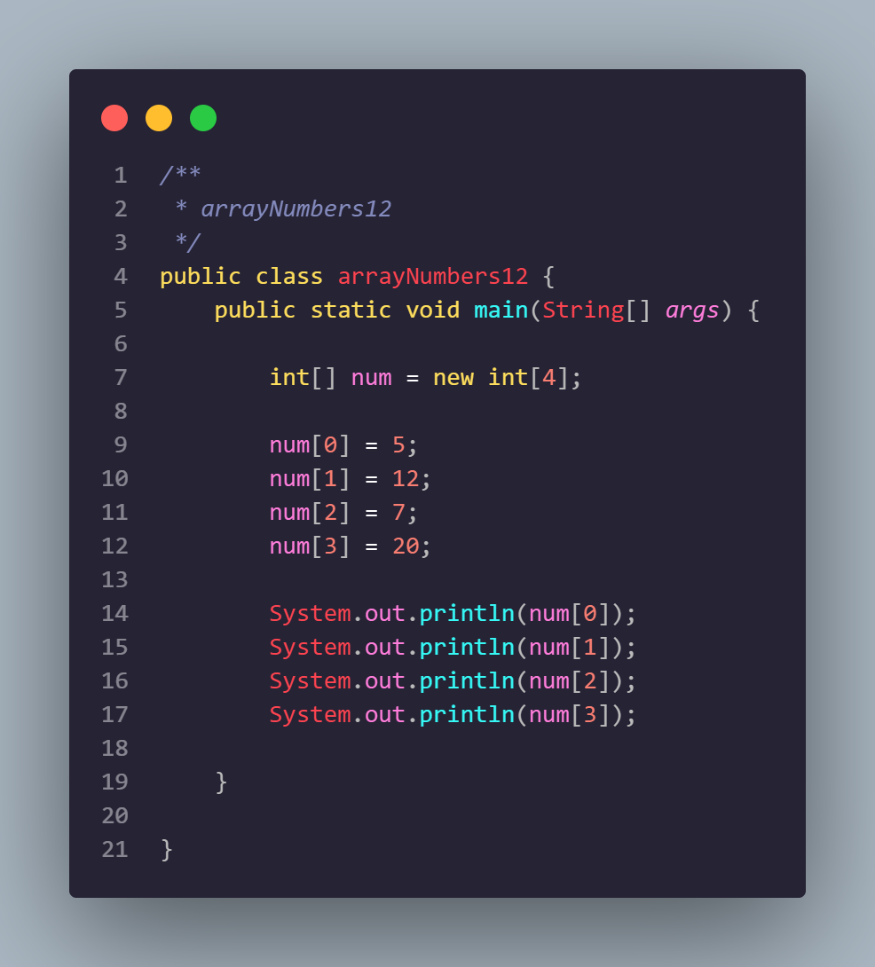
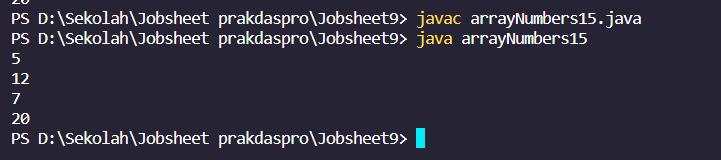
Moch. Naufal Ardian Ramadhan  
15  
SIB 1G

**Jobsheet 9**

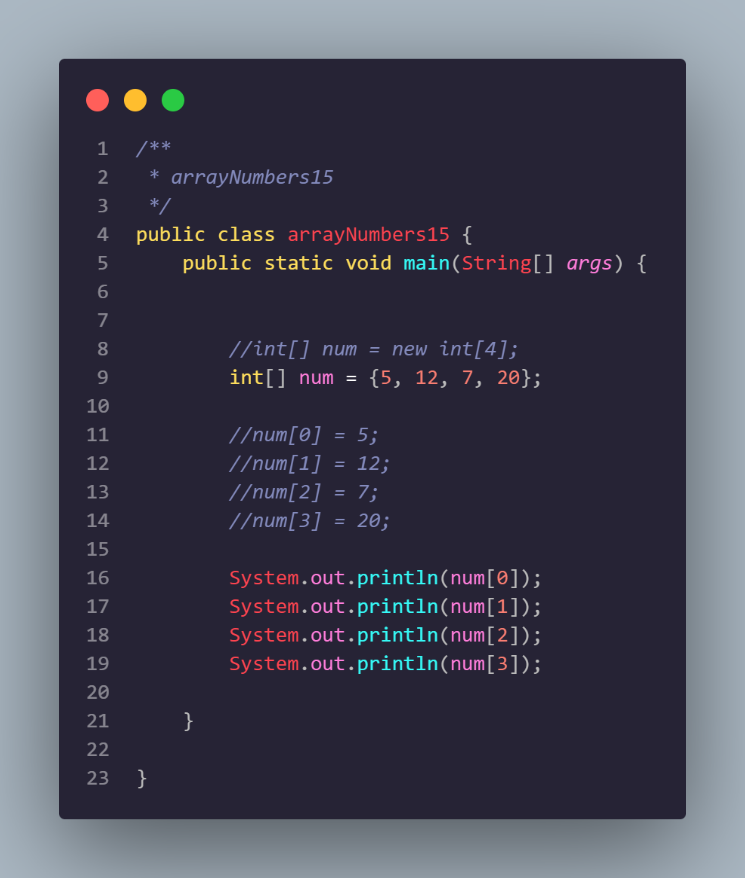
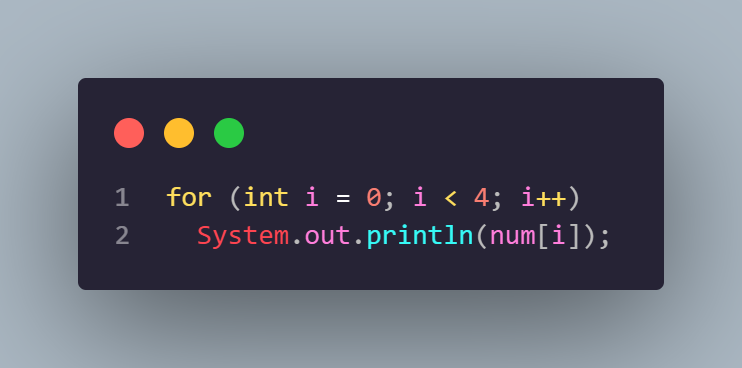
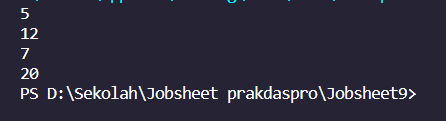
**2.1 Experiment 1: Fill in Array Element  
  
**



Question 1

1. If the contents of each element of the array num are changed with numbers 5.0, 12867, 7.5, 2000000. What happens? How can it be like that?

Cant run because the type data is int not double so cant run the program, if you want to run the program you must change from int type data to double type data

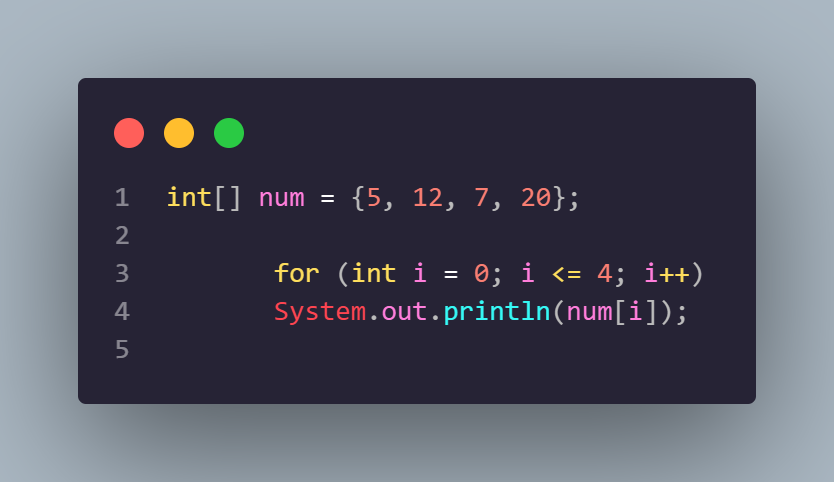
1. Modify the program code by initializing the array elements at the same time when declaring the array.  
   
2. Change the statement in step 6 to be like this  
     
   What is the result? How can it be like that?  
     
     
   The array num is declared and initialized with values {5, 12, 7, 20}.

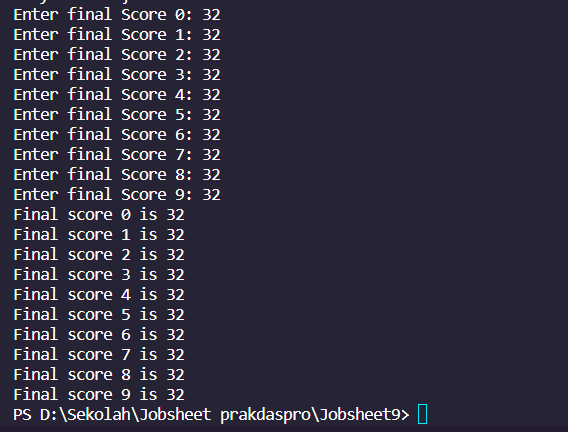
The for-loop runs from i = 0 to i < 4, and in each iteration, it prints the value of num[i].

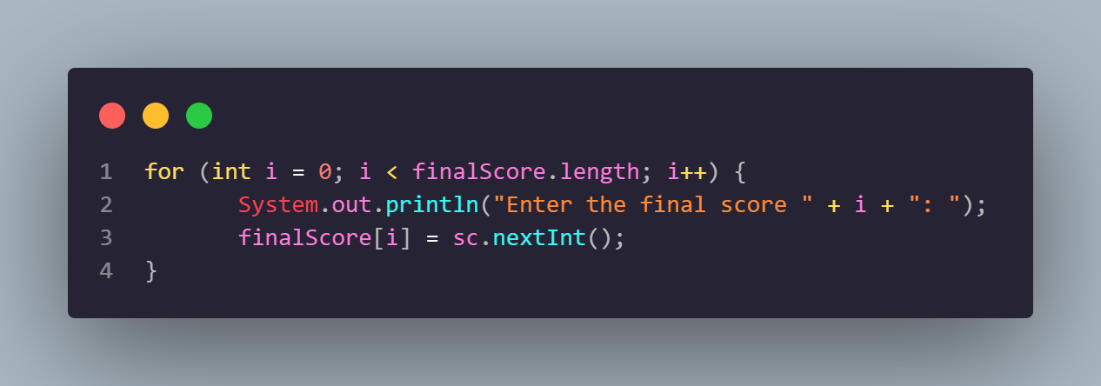
The loop will run four times (for i values 0, 1, 2, and 3), printing the elements at indices 0, 1, 2, and 3 of the array num.

The output of the program will be the values at these indices: 5, 12, 7, and 20.

This program effectively prints each element of the array on a new line, demonstrating how to access and display the values stored in an array.

1. If the condition in the for-loop statement is changed to i <= 4, what is the output of the program? Why is the result like that?  
     
     
   When the loop tries to access num[4] with i <= 4, it goes out of the bounds of the array, resulting in the mentioned exception. The correct condition to iterate over all elements in the array would be i < 4 or i < num.length.

**2.2 Experiment 2: Requesting User Input to Fill in an Array Element  
  
  
  
Result  
**

**1. Change the statement in step 5 to be like this  
**

**Run the program. Have there been any changes? How can it be like that?**In terms of functionality, the program will still prompt the user to enter final scores for 10 students, and then it will display those scores. The output and overall behavior of the program will remain the same. The change primarily improves the code's maintainability and readability.  
  
**2. Apa yang dimaksud dengan kondisi i < finalScore.length?**

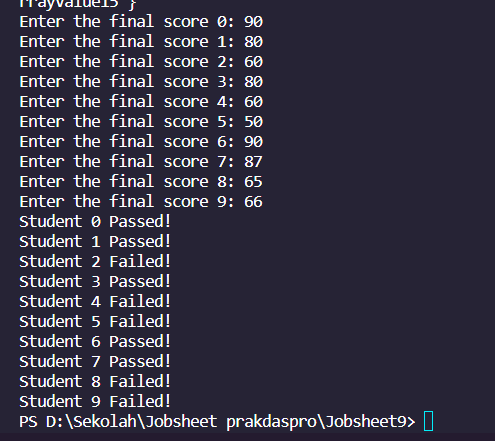
i < finalScore.length is a condition in a loop statement that means "while the value of i is less than the length of the array finalScore."

In the given program context, this statement is used in a loop to iterate through the elements of the finalScore array. Specifically, finalScore.length provides the length (the number of elements) of the array finalScore. By using the condition i < finalScore.length, it ensures that the loop will run for as many elements as there are in the finalScore array.

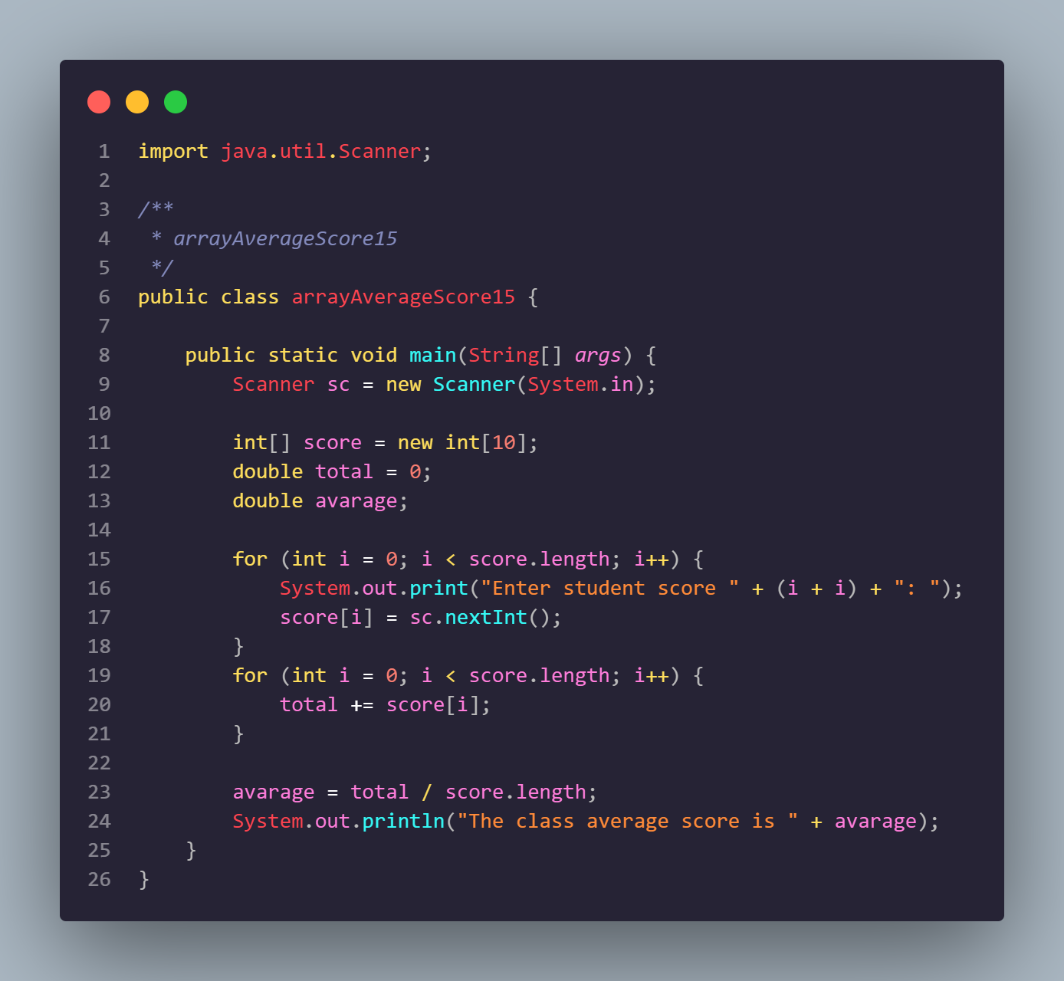
So, each element in the array will be accessed sequentially from index 0 to finalScore.length - 1 during the loop iteration. This ensures that the loop will function correctly regardless of the length of the finalScore array that may be present in the future.

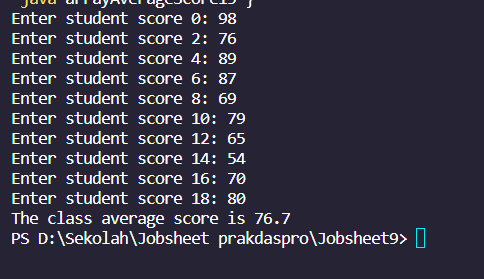
**3. Change the statement in step 6 to be like this, so that the program only displays the grades of students who passed, students who have a score > 70  
  
**

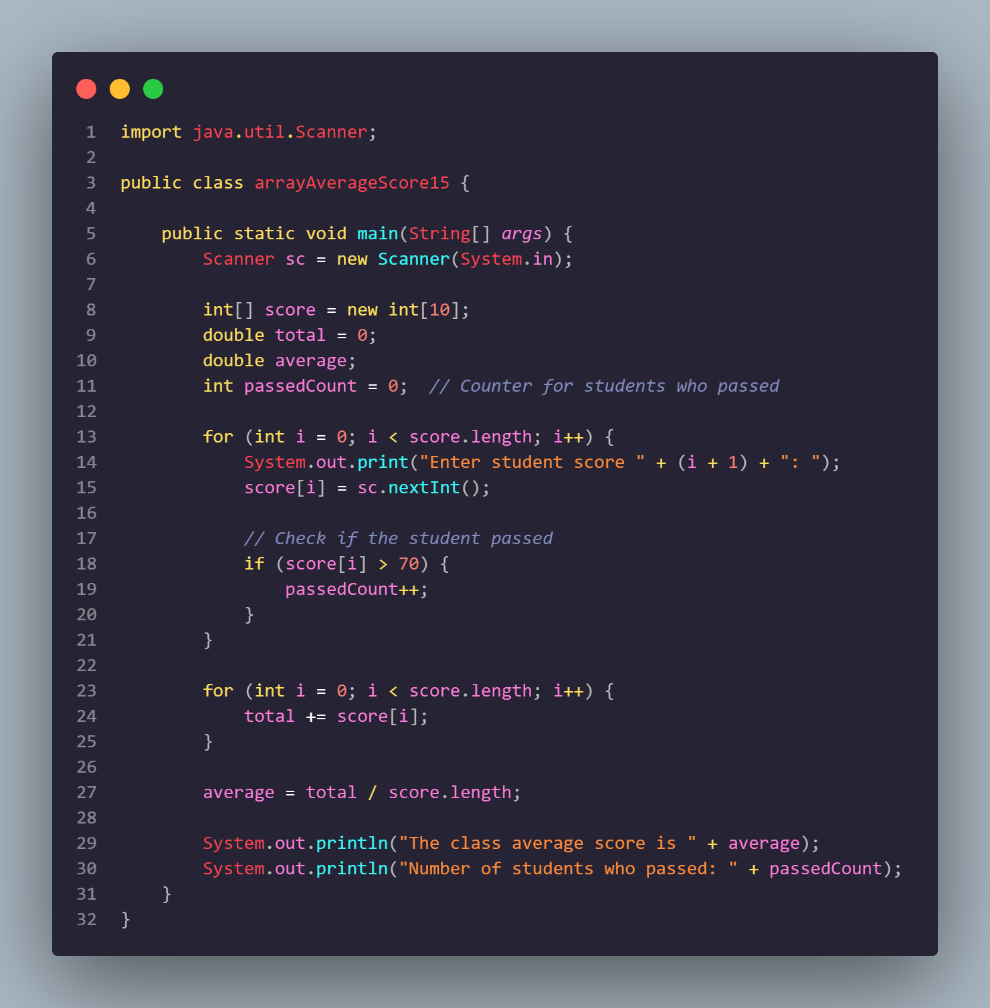
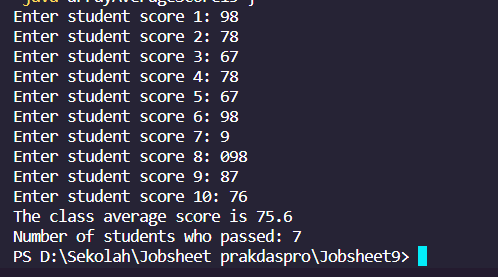
The program enters a for loop that iterates over the indices of the finalScore array.  
  
Inside the loop, there's an if statement that checks whether the value of finalScore[i] (the final score for the current student) is greater than 70.  
  
If the condition in the if statement is true, the program executes the code inside the if block. It prints a message indicating that the student at index i has passed.  
  
The loop continues to the next iteration, repeating the process for the next student.  
  
The program prints messages for each student who has a final score greater than 70, indicating that they have passed.

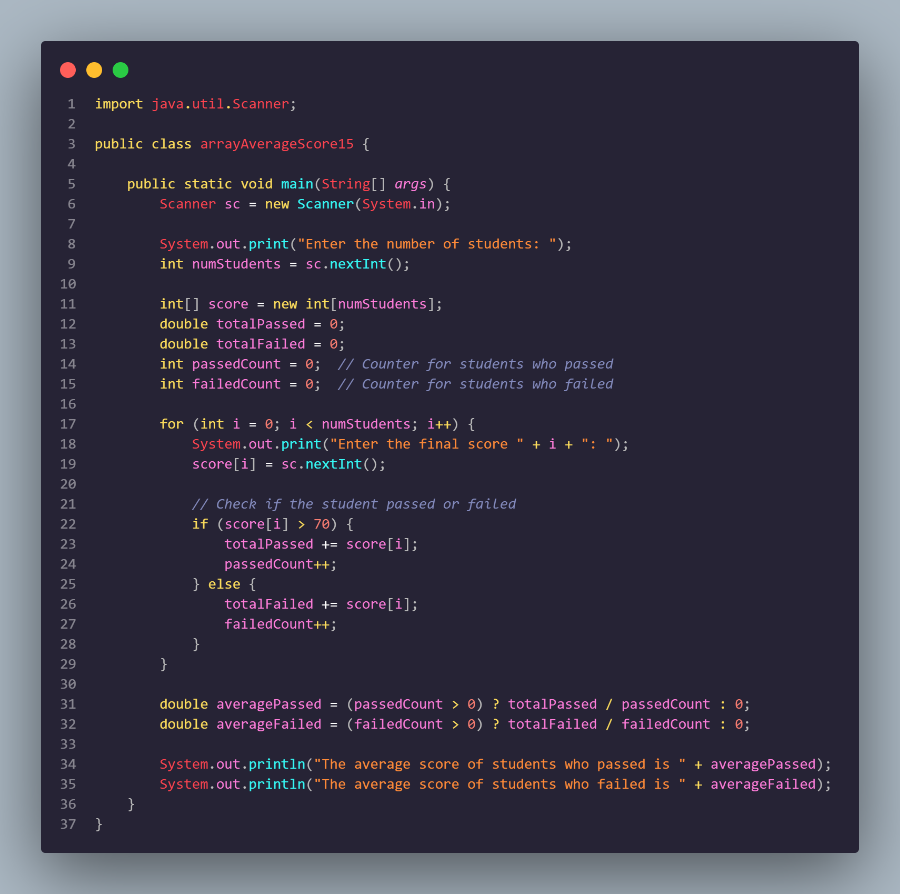
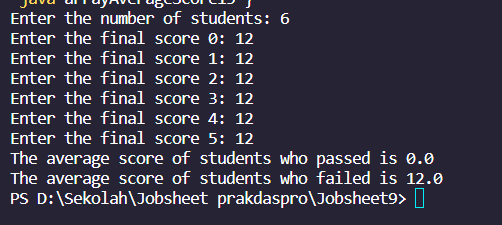
**4. Modify the program so that it displays all students, and mark which one passed, and which did not!** **Code  
**

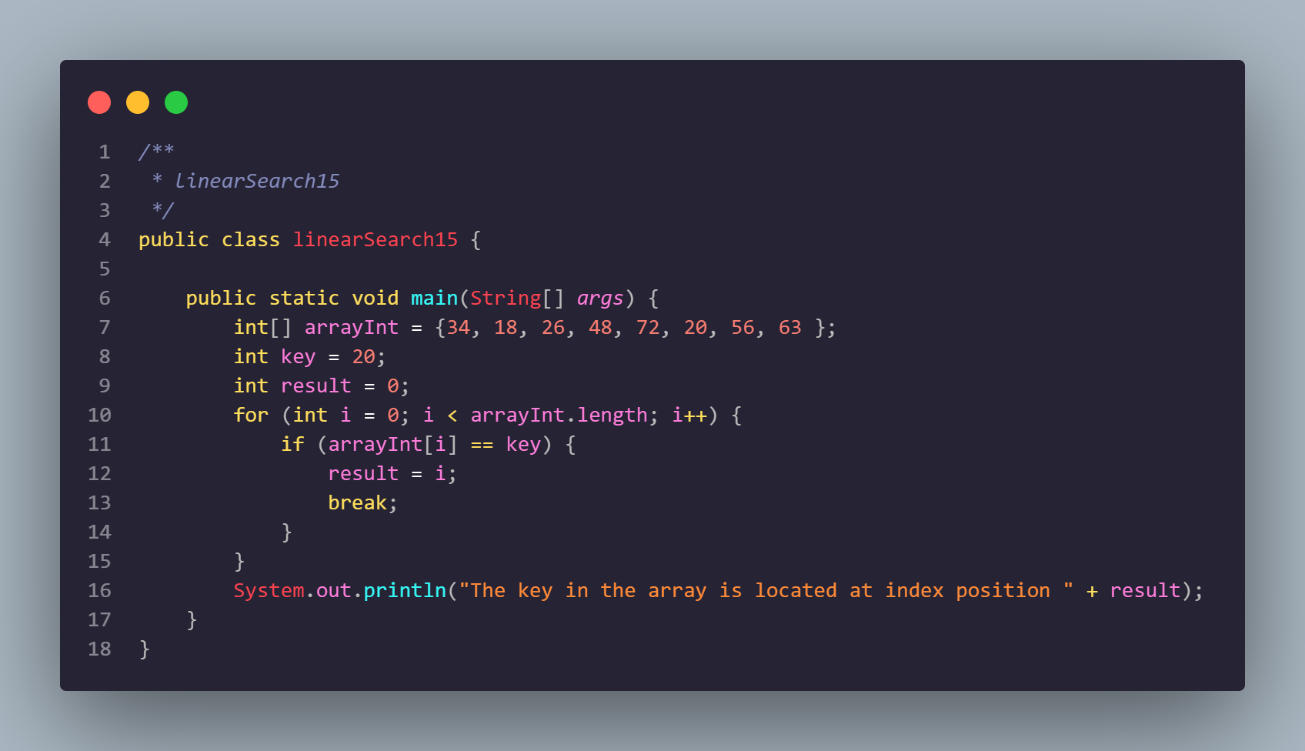
**2.3 Experiment 3: Perform Arithmetic Operations on Array Elements**

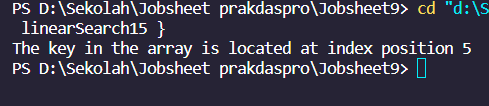
**Experiment Time: 75 minutes  
  
**

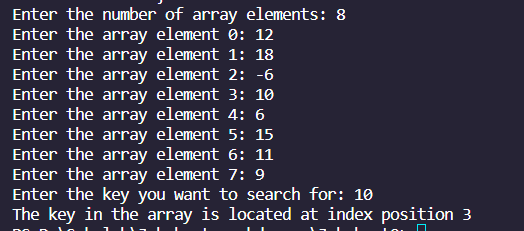
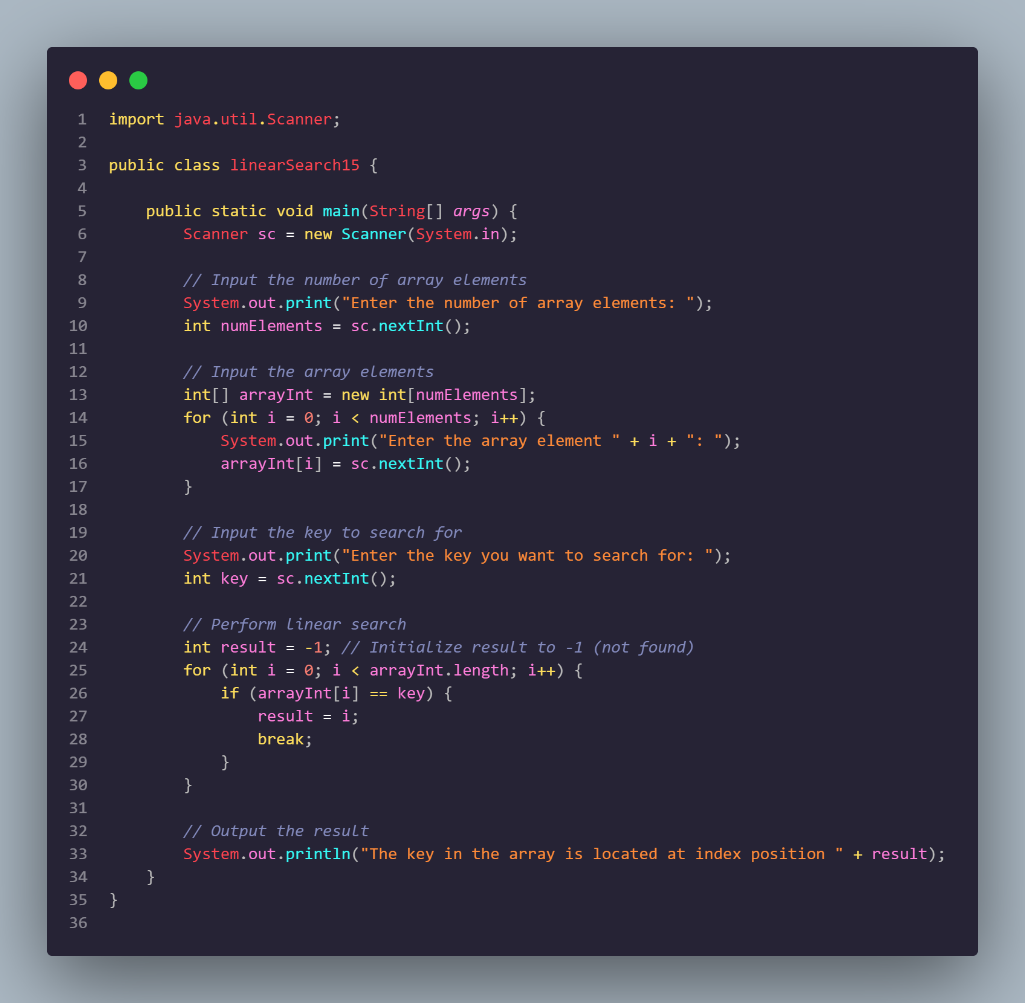
**Terminal  
**

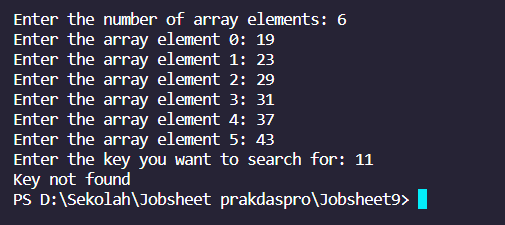
**1. Modify the program in Experiment 3 so that the program can display the number of students who passed, students who have a score greater than 70 (>70)  
  
  
  
Terminal  
**

**2. Modify the program in Experiment 3 so that it can produce output like the following display  
  
  
  
Terminal  
  
**

**2.4 Experiment 4: Searching  
  
**

**Terminal  
  
  
1. Explain the meaning of the break; statement on line 13 of the program code in Experiment 4**  
The `break;` statement is used to exit a loop prematurely when a certain condition is met. In the provided code, it is used to terminate the loop as soon as the key is found in the array, preventing unnecessary iterations after the desired element has been located.

**2. Modify the program code in experiment 4 so that the program can receive input in the form of the number of array elements, the contents of the array, and the key you want to search for. Then, print to the screen the index of the element positions of the searched key. Example of program results:  
  
  
  
Code  
**

**3. Modify the program in experiment 4 so that the program will give the message "key not found" if the key is not in the array. Example of program results:  
  
  
  
Code  
  
**

**3. Assignment**

**1. Create a program to produce the highest value, lowest value, and average from an array containing integer type numbers.**

**Terms:**

**• Input: Number of elements, value of each element**

**• Output: Highest value, lowest value, average value**

**Programming Fundamentals 2023**

**2. Implement the flowchart that was created in the assignment for Week 9 of the**

**Programming Fundamentals course related to the group project into Java program**

**code.**

**Commit and push the results of your program code to your project's GitHub repository.**

**Note: assignments may only apply material from Week 1 to Week 9.**