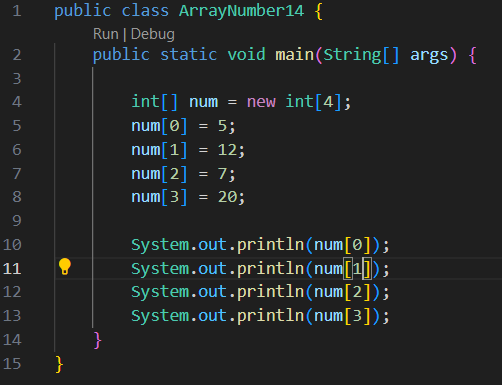
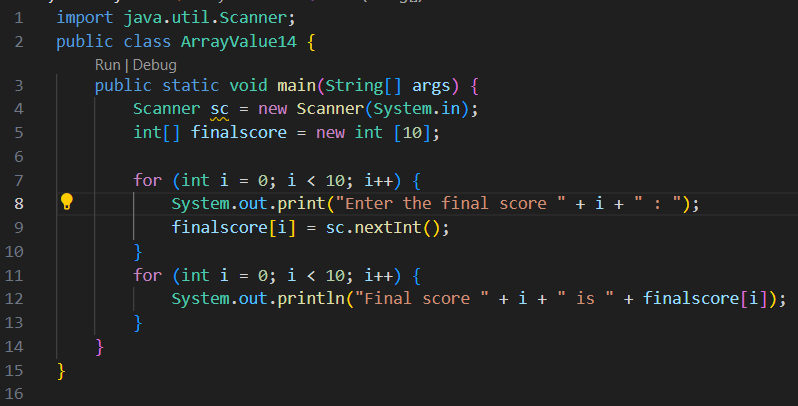
|  |  |  |
| --- | --- | --- |
| **NAME** | **:** | **MUCHAMMAD NABIL HAYKAL WIDARTO** |
| **CLASS** | **:** | **SIB-1G** |
| **NIM** | **:** | **2341760152** |

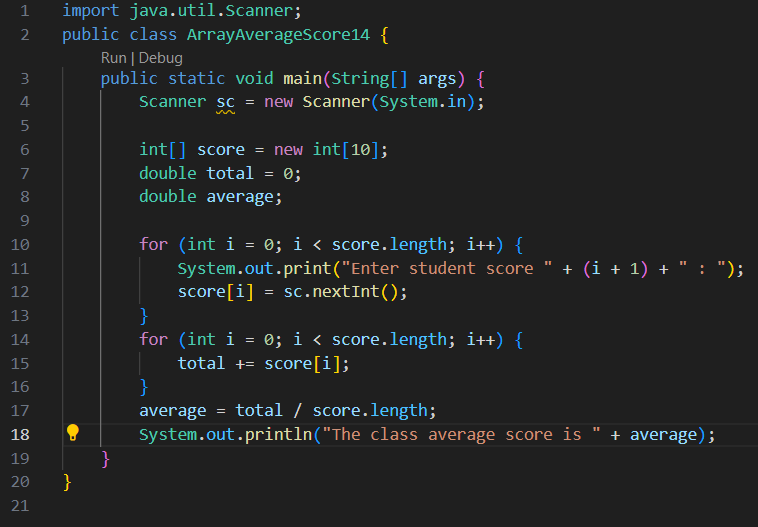
**JOBSHEET 9**

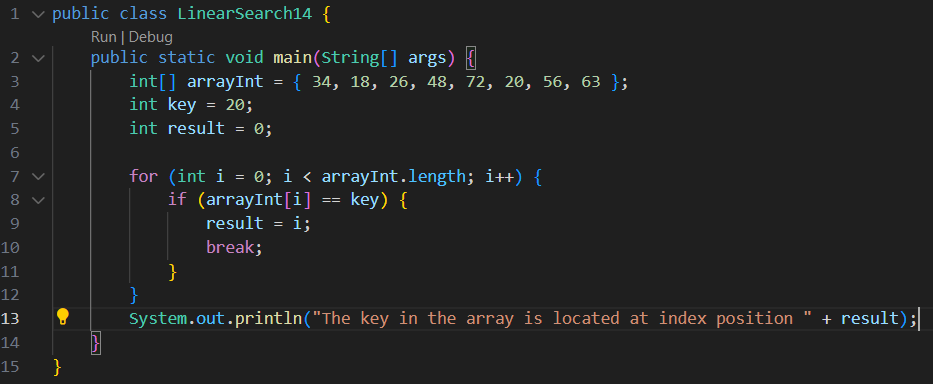
**https://github.com/muchnabil/daspro-jobsheet9**

**EXPERIMENT**

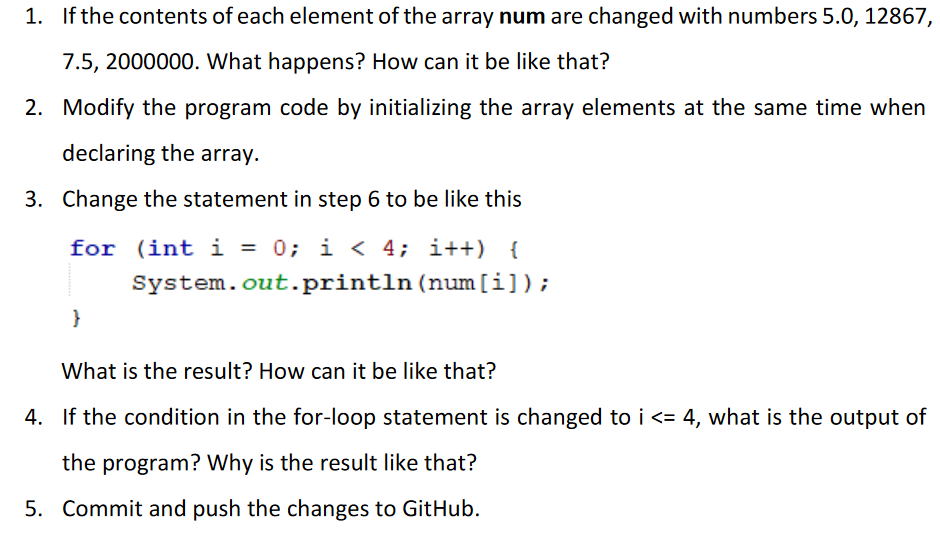
1. 

2. 

3. 

4. 

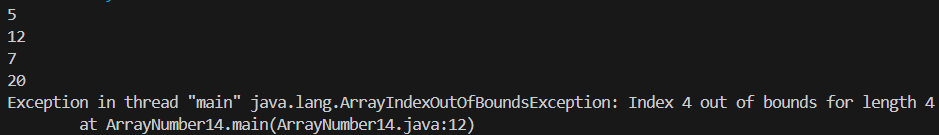
QUESTION 2.1



**ANSWER**

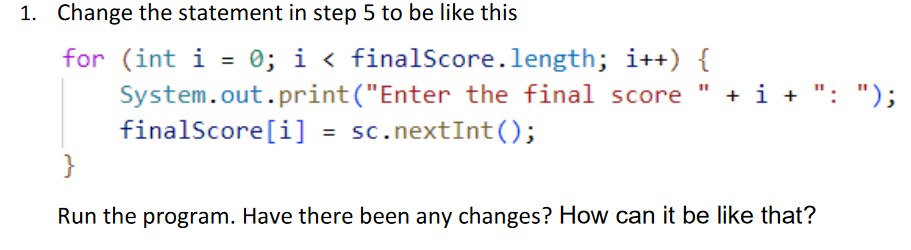
1. Error because the data type used is int[], while the value to be entered must use the double[] data type.

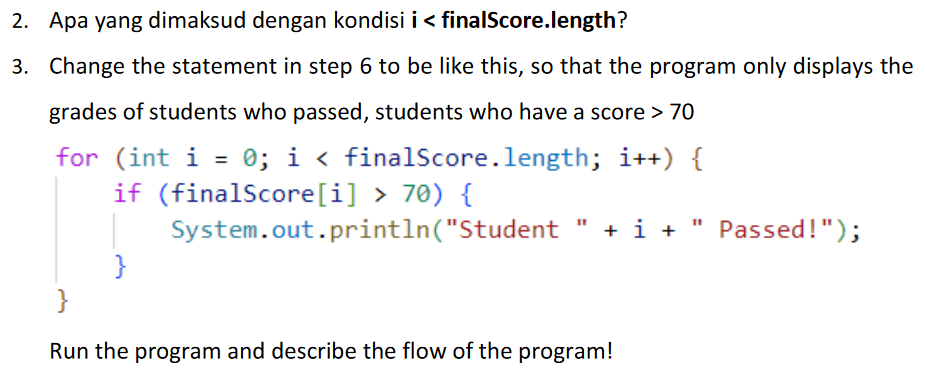
3.  the result is the same, but the process is different, this process uses looping to make the program shorter

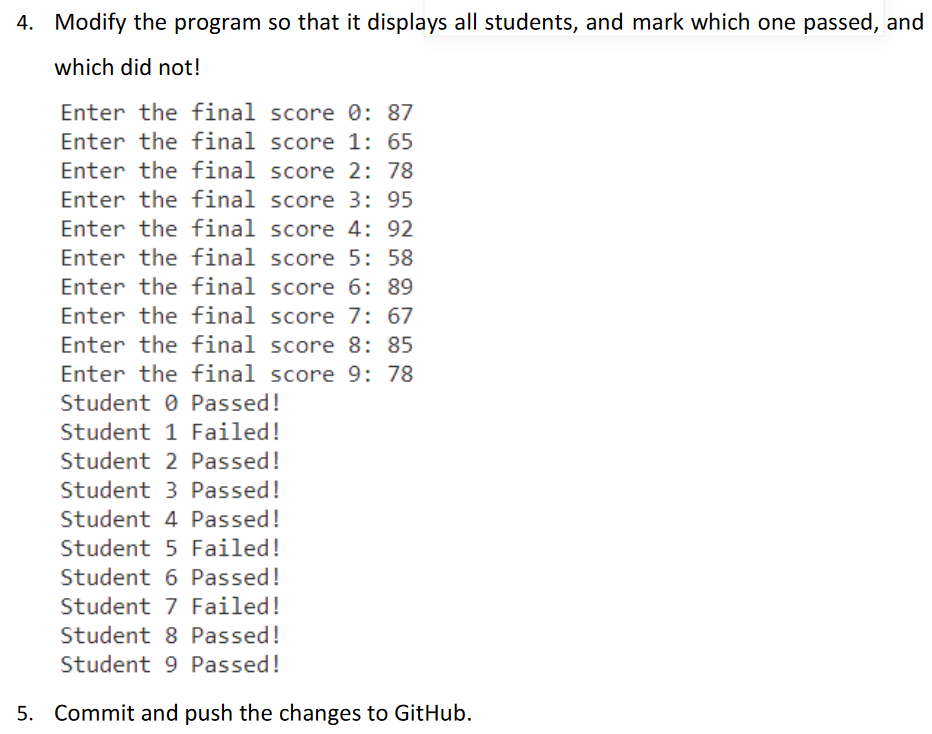
4. 

error results because the declared limit is only 4.

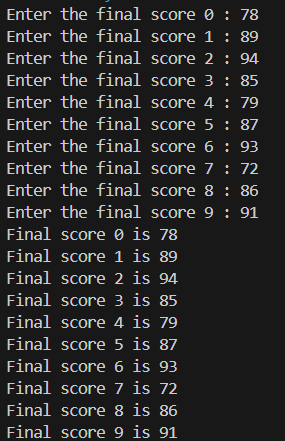
**QUESTION 2.2**





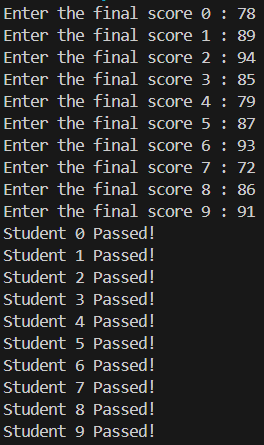


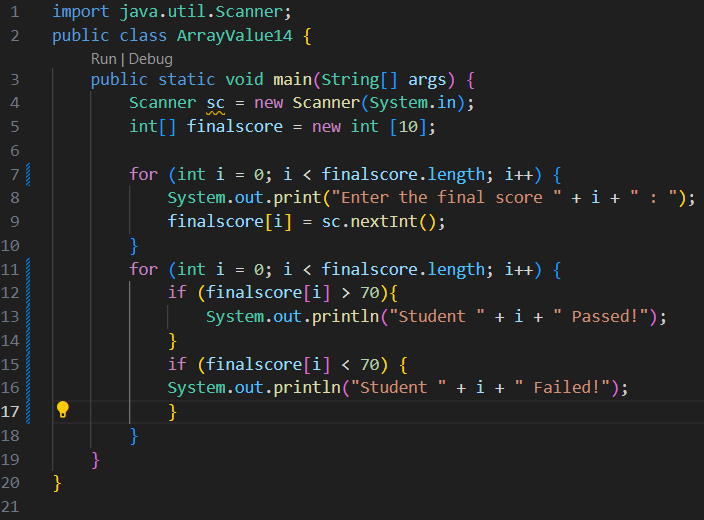
**ANSWER**

1. 

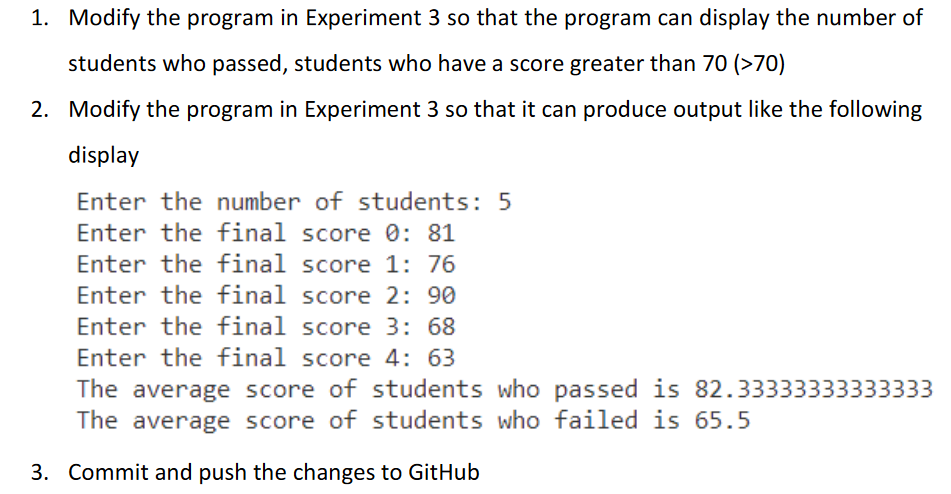
The results are the same but the method is shorter and easier to understand.

2. *finalScore.length* is a property in Java that gives the length (number of elements) of the finalScore array. So, if finalScore contains 5 elements, *finalScore.length* will be 5

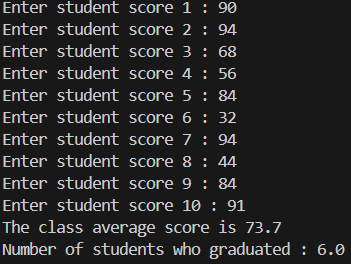
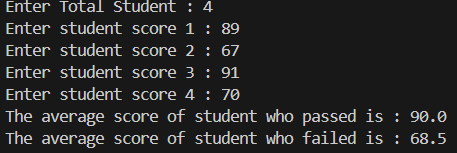
3.  The final score of all grades is declared passed because the passing limit is > 70

4. 

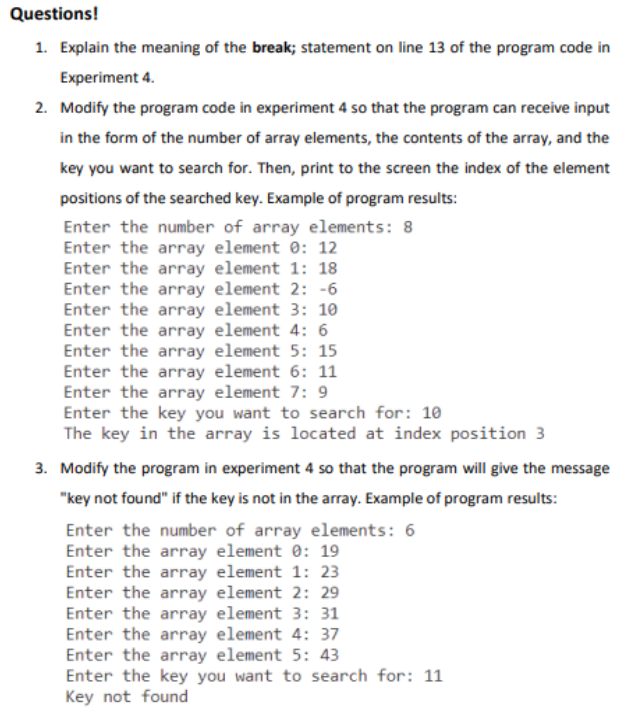
**QUESTION 2.3**

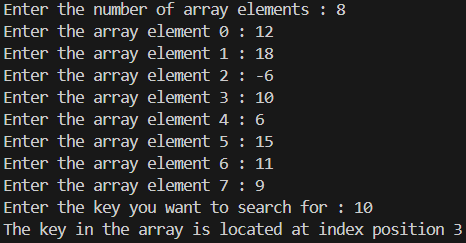
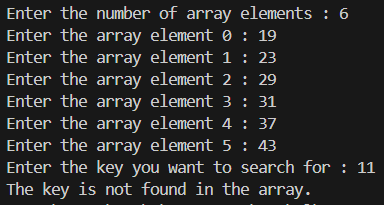


**ANSWER**

1. 
2. 

**Question 2.4**



1. break; statement on line 13 serves to stop the search or loop after the searched value is found. For example, here the key is 20, the program will continue to run the loop until the compared value is equal to the the value of key or 20, when the value is the same the break statement will be executed, so the the loop stops.
2. 
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

1. 

