

NAME – RAJDEEP JAISWAL

DATE – 08 Oct 2021

BRANCH – BTECH CSE

SEC = 13 A

UID -20BCS2761

SUB- JAVA ASSIGNMENT

Q 5 - Write a program to input n numbers on command line argument and calculate maximum of them.

CODES –

```
class Max
{
    public static void main(String[] args)
    {
        if (args.length == 0)
        {
            System.out.println("You must enter at least 1 integer. Please try again.");
            System.exit(0);
        }
        findMax(args);
    }

    static void findMax(String[] strValues)
    {
        int numMax = Integer.parseInt(strValues[0]);
        //int maxValue=0;
        for (int i = 0; i < strValues.length; i++)
        {
            int numbers = Integer.parseInt(strValues[i]);
            if(numbers[i] > numMax){
                numMax = numbers[i];
                System.out.println(numMax);
            }
        }
    }
}
```

```
        System.out.println(numMax);
    }
}
}
```

OUTPUT -

7

8

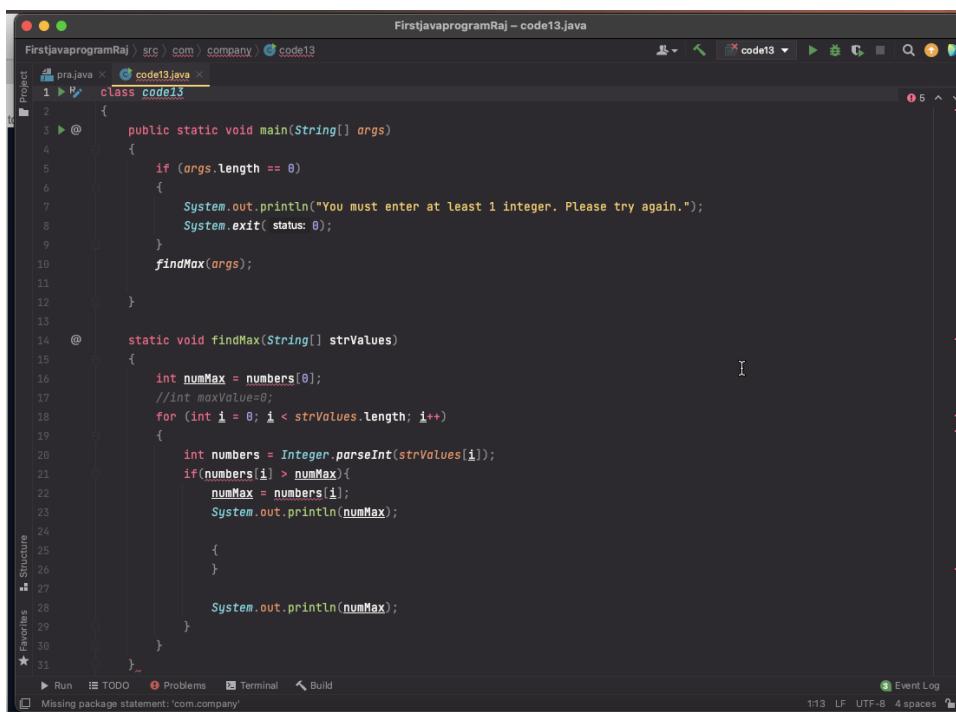
80

3

2

7

CODE in compiler -



The screenshot shows an IDE interface with the following details:

- Title Bar:** FirstjavaprogramRaj – code13.java
- Project Explorer:** Shows a project structure with a file named code13.java.
- Code Editor:** Displays the Java code for the class code13. The code includes a main method that checks if there are at least one argument. If not, it prints a message and exits. Otherwise, it calls the static method findMax with the arguments. The findMax method initializes numMax to the first argument, then iterates through the remaining arguments, updating numMax if a larger value is found. Finally, it prints the maximum value.
- Status Bar:** Shows "Missing package statement: 'com.company'" and other build-related information.

```
1  public class code13 {
2      public static void main(String[] args) {
3          if (args.length == 0) {
4              System.out.println("You must enter at least 1 integer. Please try again.");
5              System.exit( status: 0 );
6          }
7          findMax(args);
8      }
9
10     @
11     static void findMax(String[] strValues) {
12         int numMax = numbers[0];
13         //int maxValue=0;
14         for (int i = 0; i < strValues.length; i++) {
15             int numbers = Integer.parseInt(strValues[i]);
16             if(numbers[i] > numMax){
17                 numMax = numbers[i];
18                 System.out.println(numMax);
19             }
20         }
21         System.out.println(numMax);
22     }
23     }
24
25
26
27
28
29
30
31 }
```