Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

LAB NO 1

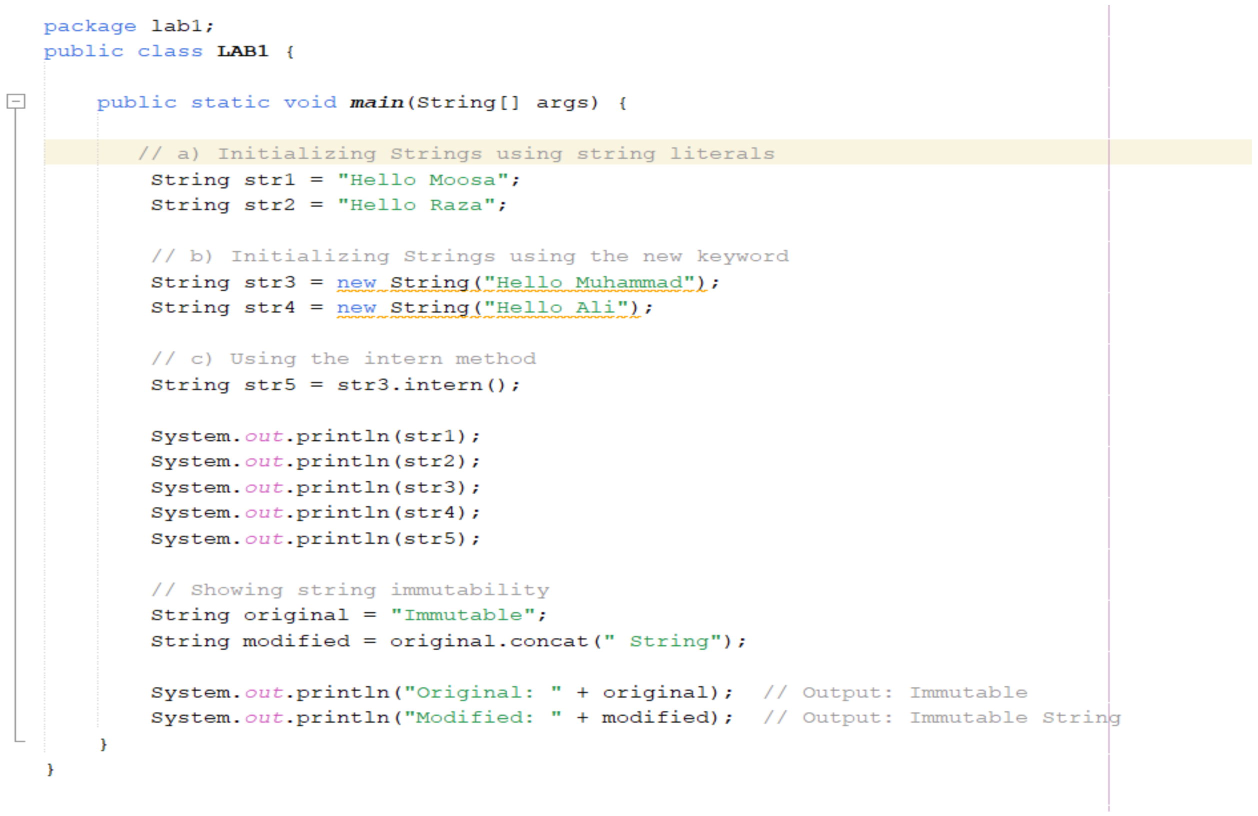
DATA STRUCTURES AND ALGORITHMS

OBJECTIVE: To study the concepts of String Constant Pool, String literals, String immutability and Wrapper classes.

TASK NO 1:

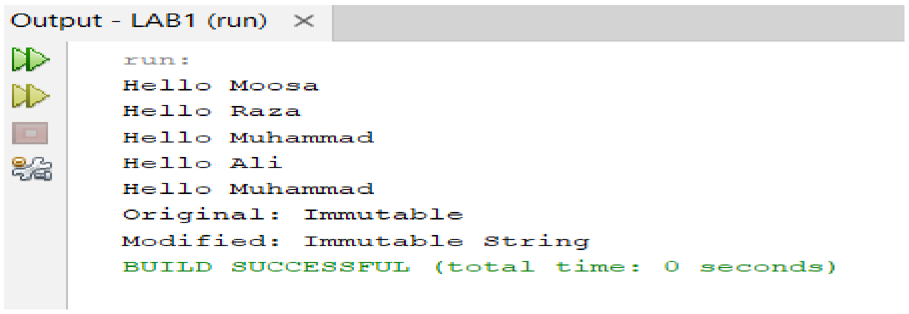
Write a program that initialize five different strings using all the above mentioned ways, i.e., a)string literals b)new keyword also use intern method and show string immutability.

INPUT:



```
package lab1;
public class LAB1 {

    public static void main(String[] args) {

        // a) Initializing Strings using string literals
        String str1 = "Hello Moosa";
        String str2 = "Hello Raza";

        // b) Initializing Strings using the new keyword
        String str3 = new String("Hello Muhammad");
        String str4 = new String("Hello Ali");

        // c) Using the intern method
        String str5 = str3.intern();

        System.out.println(str1);
        System.out.println(str2);
        System.out.println(str3);
        System.out.println(str4);
        System.out.println(str5);

        // Showing string immutability
        String original = "Immutable";
        String modified = original.concat(" String");

        System.out.println("Original: " + original);  // Output: Immutable
        System.out.println("Modified: " + modified);  // Output: Immutable String
    }
}
```

OUTPUT:



Output - LAB1 (run)

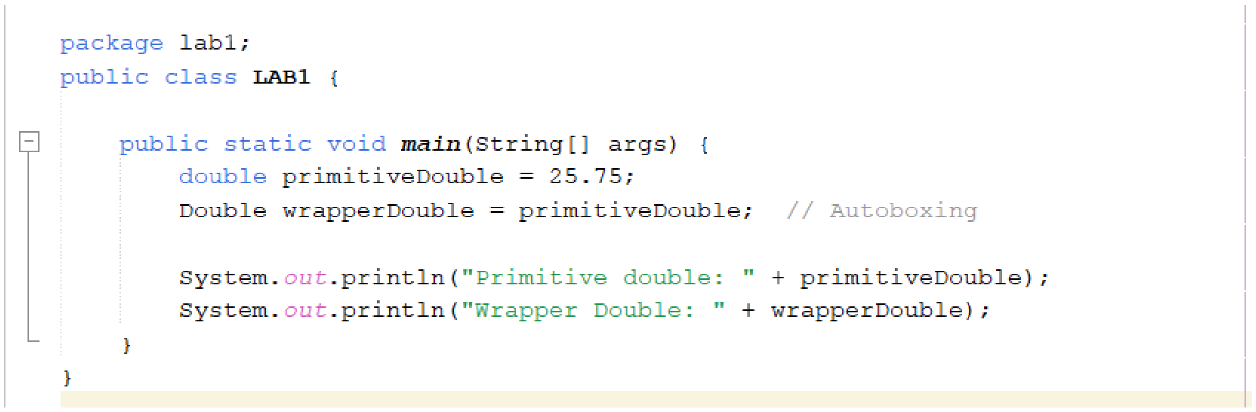
```
run:
Hello Moosa
Hello Raza
Hello Muhammad
Hello Ali
Hello Muhammad
Original: Immutable
Modified: Immutable String
BUILD SUCCESSFUL (total time: 0 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 2:

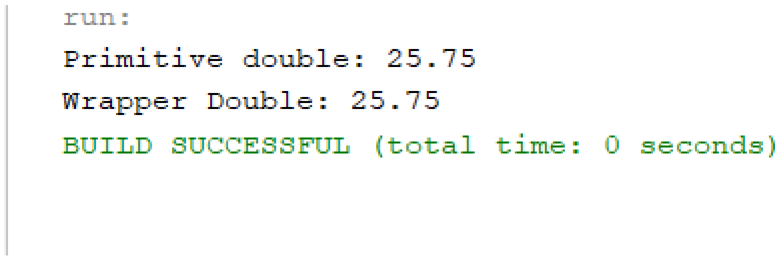
Write a program to convert primitive data type Double into its respective wrapper object.

INPUT:



```java
package lab1;
public class LAB1 {

    public static void main(String[] args) {
        double primitiveDouble = 25.75;
        Double wrapperDouble = primitiveDouble;  // Autoboxing

        System.out.println("Primitive double: " + primitiveDouble);
        System.out.println("Wrapper Double: " + wrapperDouble);
    }
}
```

OUTPUT:



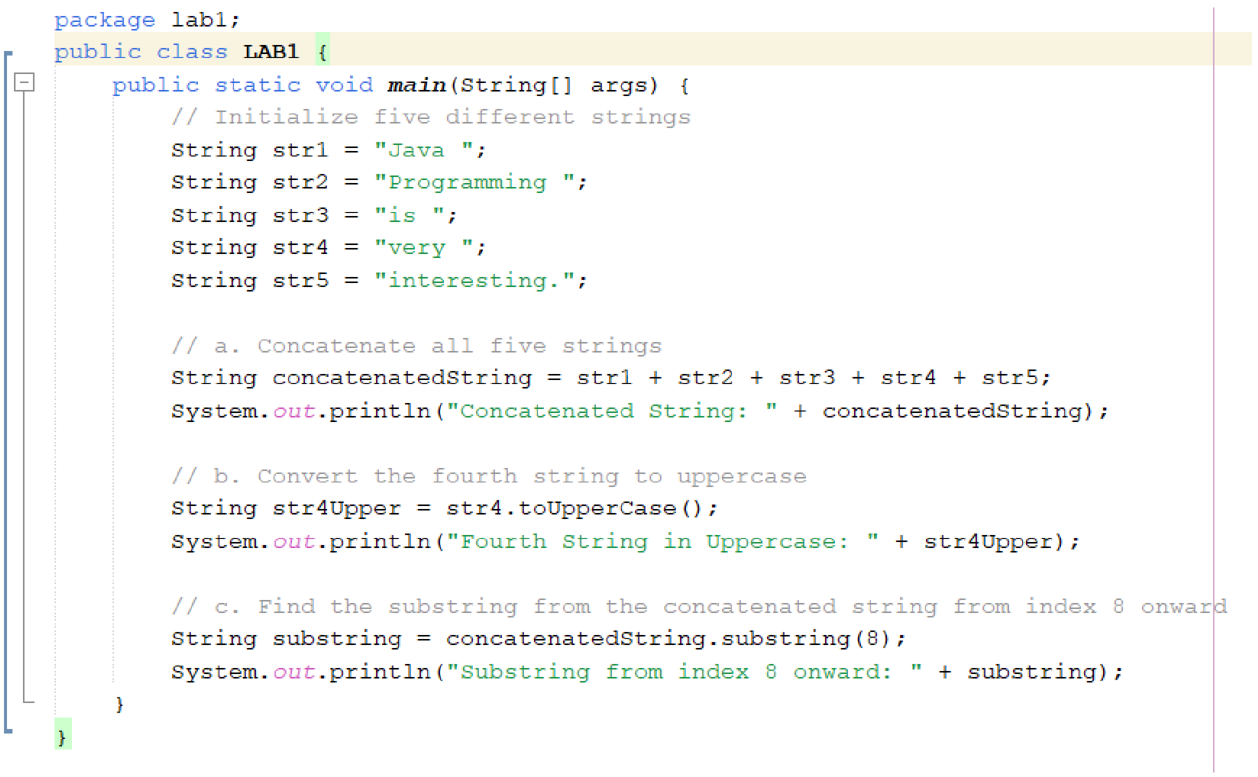
```
run:
Primitive double: 25.75
Wrapper Double: 25.75
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 3:

Write a program that initialize five different strings and perform the following operations. a. Concatenate all five stings. b. Convert fourth string to uppercase. c. Find the substring from the concatenated string from 8 to onward.

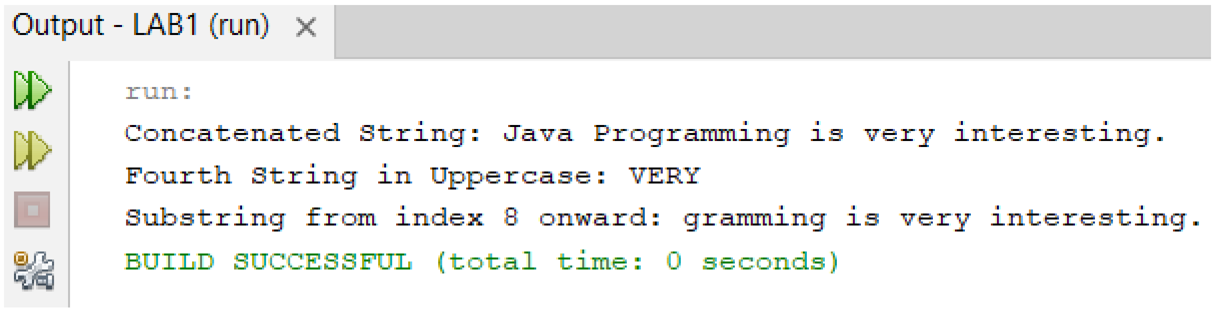
Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

INPUT:



```
package lab1;
public class LAB1 {
    public static void main(String[] args) {
        // Initialize five different strings
        String str1 = "Java ";
        String str2 = "Programming ";
        String str3 = "is ";
        String str4 = "very ";
        String str5 = "interesting.";

        // a. Concatenate all five strings
        String concatenatedString = str1 + str2 + str3 + str4 + str5;
        System.out.println("Concatenated String: " + concatenatedString);

        // b. Convert the fourth string to uppercase
        String str4Upper = str4.toUpperCase();
        System.out.println("Fourth String in Uppercase: " + str4Upper);

        // c. Find the substring from the concatenated string from index 8 onward
        String substring = concatenatedString.substring(8);
        System.out.println("Substring from index 8 onward: " + substring);
    }
}
```

OUTPUT:



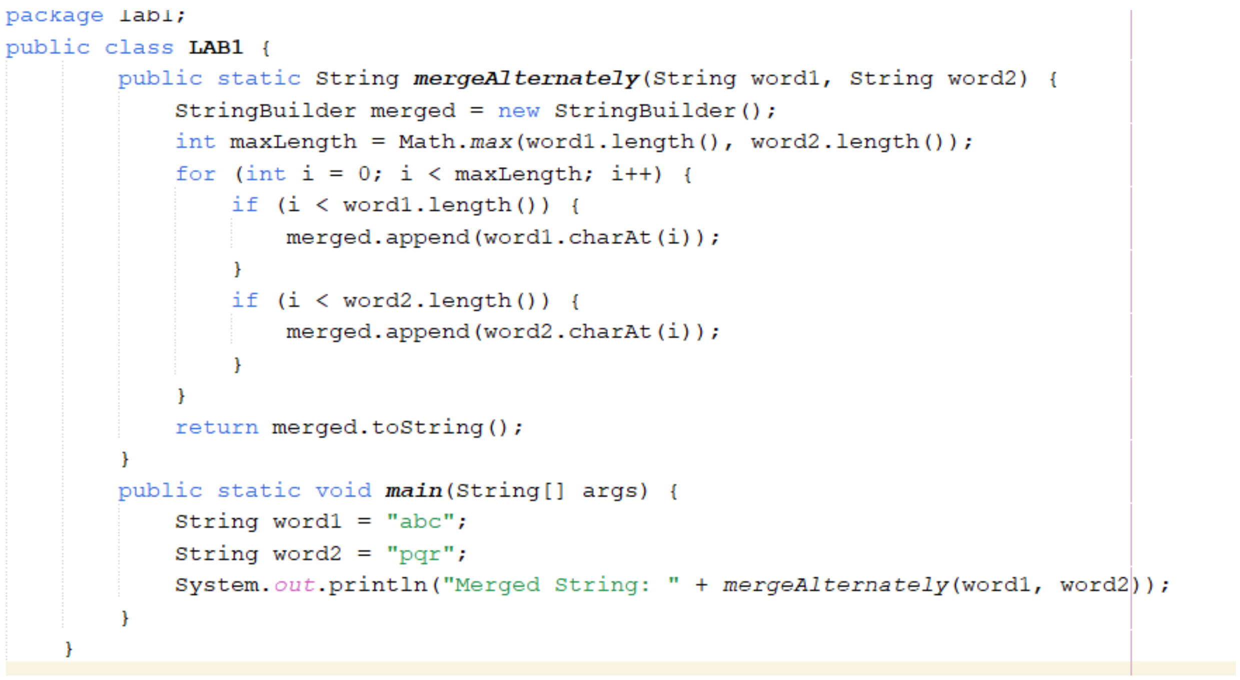
```
run:
Concatenated String: Java Programming is very interesting.
Fourth String in Uppercase: VERY
Substring from index 8 onward: gramming is very interesting.
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 4:

You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string. Return the merged string. Example: Input: word1 = "abc", word2 = "pqr" Output: "apbqcr" Explanation: The merged string will be merged as so: word1: a b c word2: p q r merged: a p b q c r

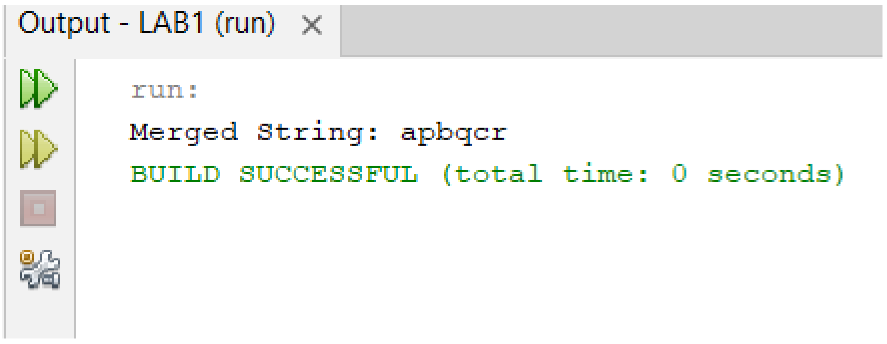
Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

INPUT:



```java
package lab1;
public class LAB1 {
        public static String mergeAlternately(String word1, String word2) {
            StringBuilder merged = new StringBuilder();
            int maxLength = Math.max(word1.length(), word2.length());
            for (int i = 0; i < maxLength; i++) {
                if (i < word1.length()) {
                    merged.append(word1.charAt(i));
                }
                if (i < word2.length()) {
                    merged.append(word2.charAt(i));
                }
            }
            return merged.toString();
        }
        public static void main(String[] args) {
            String word1 = "abc";
            String word2 = "pqr";
            System.out.println("Merged String: " + mergeAlternately(word1, word2));
        }
    }
```

OUTPUT:



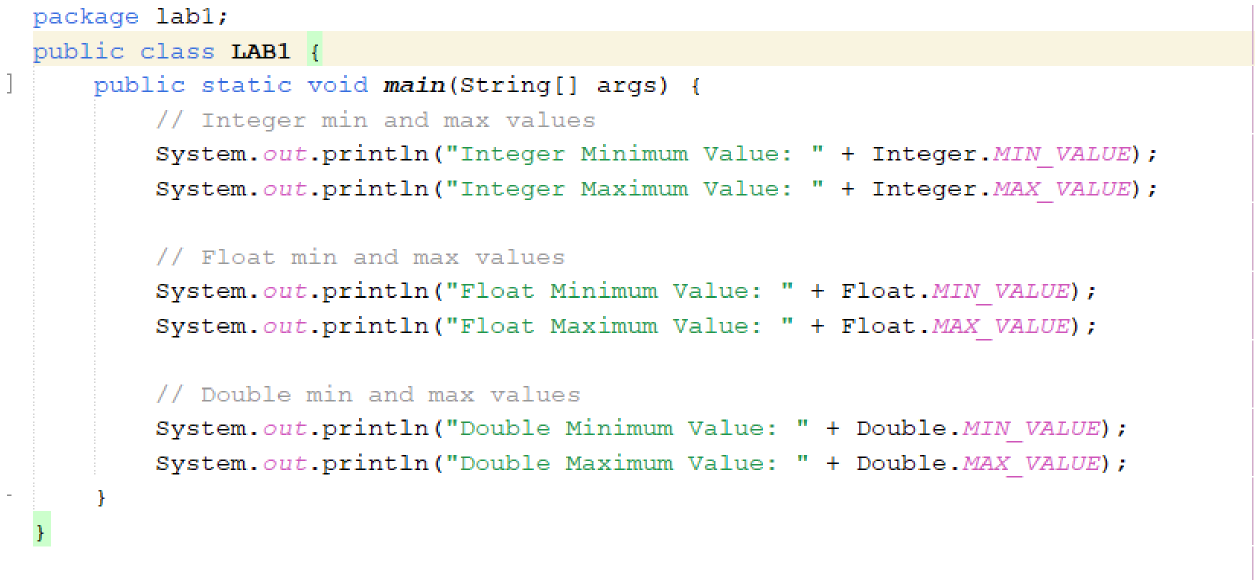
```
run:
Merged String: apbqcr
BUILD SUCCESSFUL (total time: 0 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 5:

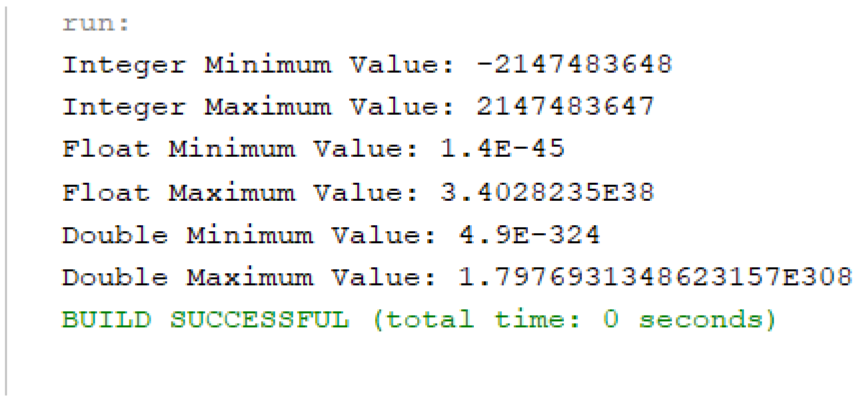
Write a Java program to find the minimum and maximum values of Integer, Float, and Double using the respective wrapper class constants.

INPUT:



```java
package lab1;
public class LAB1 {
    public static void main(String[] args) {
        // Integer min and max values
        System.out.println("Integer Minimum Value: " + Integer.MIN_VALUE);
        System.out.println("Integer Maximum Value: " + Integer.MAX_VALUE);

        // Float min and max values
        System.out.println("Float Minimum Value: " + Float.MIN_VALUE);
        System.out.println("Float Maximum Value: " + Float.MAX_VALUE);

        // Double min and max values
        System.out.println("Double Minimum Value: " + Double.MIN_VALUE);
        System.out.println("Double Maximum Value: " + Double.MAX_VALUE);
    }
}
```

OUTPUT:



```
run:
Integer Minimum Value: -2147483648
Integer Maximum Value: 2147483647
Float Minimum Value: 1.4E-45
Float Maximum Value: 3.4028235E38
Double Minimum Value: 4.9E-324
Double Maximum Value: 1.7976931348623157E308
BUILD SUCCESSFUL (total time: 0 seconds)
```

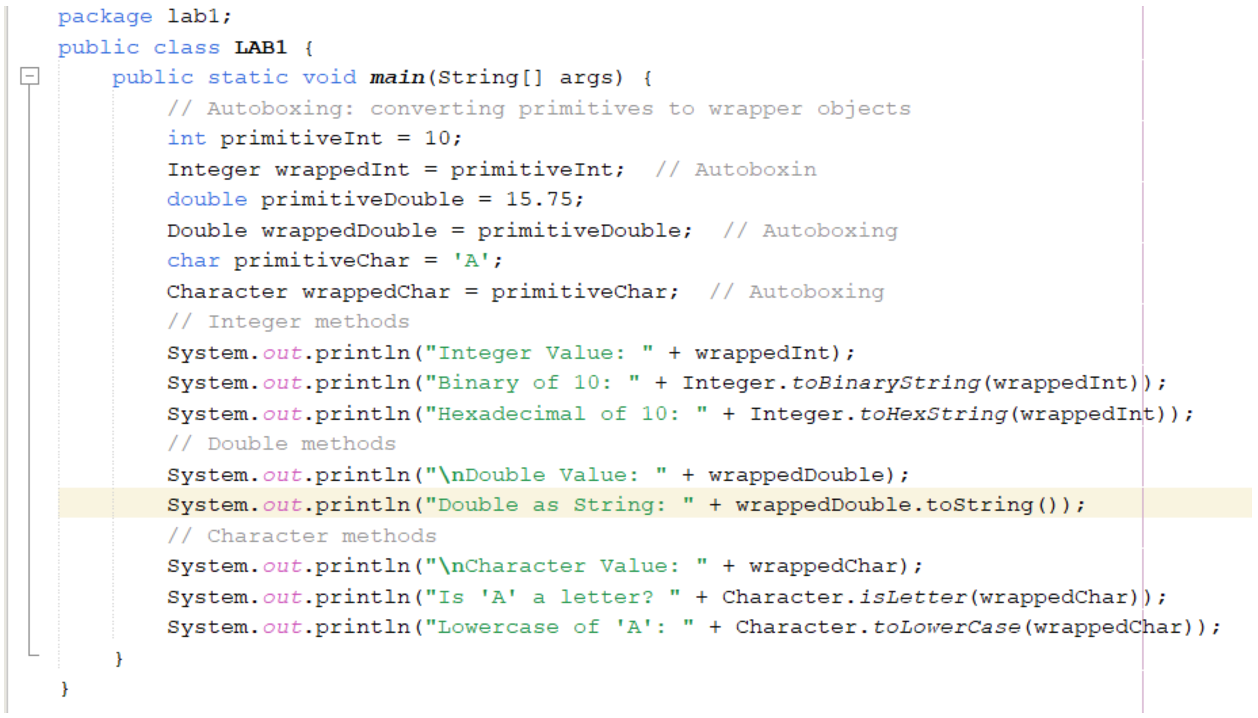
Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

HOME TASKS

TASK NO 1:

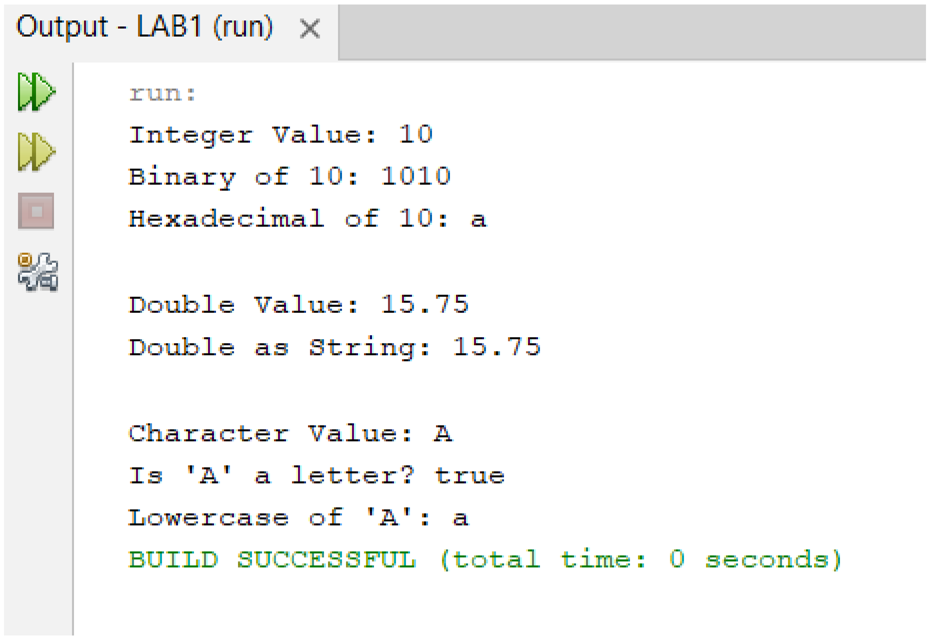
Write a JAVA program to perform Autoboxing and also implement different methods of wrapper class.

INPUT:



```
package lab1;
public class LAB1 {
    public static void main(String[] args) {
        // Autoboxing: converting primitives to wrapper objects
        int primitiveInt = 10;
        Integer wrappedInt = primitiveInt;  // Autoboxin
        double primitiveDouble = 15.75;
        Double wrappedDouble = primitiveDouble;  // Autoboxing
        char primitiveChar = 'A';
        Character wrappedChar = primitiveChar;  // Autoboxing
        // Integer methods
        System.out.println("Integer Value: " + wrappedInt);
        System.out.println("Binary of 10: " + Integer.toBinaryString(wrappedInt));
        System.out.println("Hexadecimal of 10: " + Integer.toHexString(wrappedInt));
        // Double methods
        System.out.println("\nDouble Value: " + wrappedDouble);
        System.out.println("Double as String: " + wrappedDouble.toString());
        // Character methods
        System.out.println("\nCharacter Value: " + wrappedChar);
        System.out.println("Is 'A' a letter? " + Character.isLetter(wrappedChar));
        System.out.println("Lowercase of 'A': " + Character.toLowerCase(wrappedChar));
    }
}
```

OUTPUT:



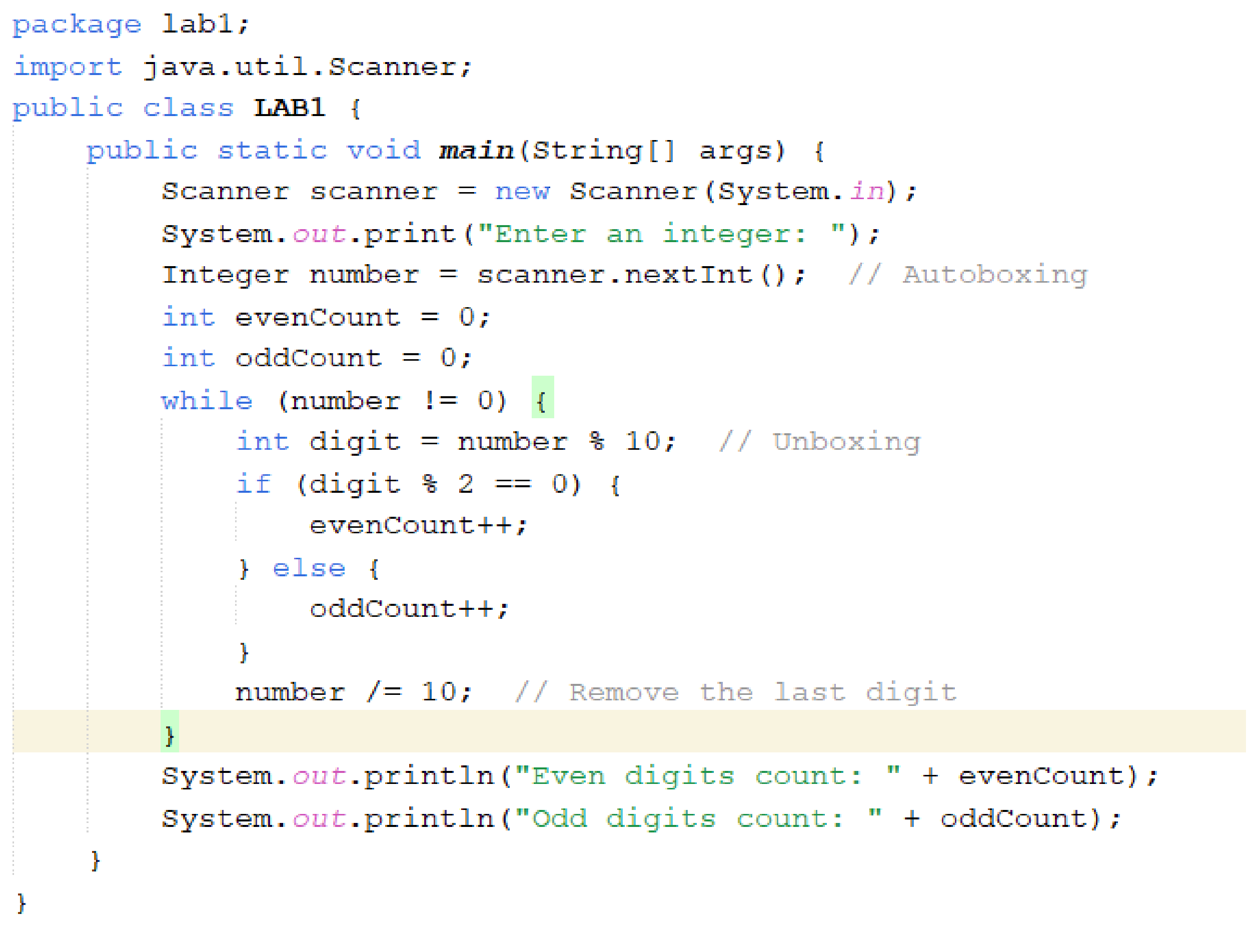
```
run:
Integer Value: 10
Binary of 10: 1010
Hexadecimal of 10: a

Double Value: 15.75
Double as String: 15.75

Character Value: A
Is 'A' a letter? true
Lowercase of 'A': a
BUILD SUCCESSFUL (total time: 0 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 2:

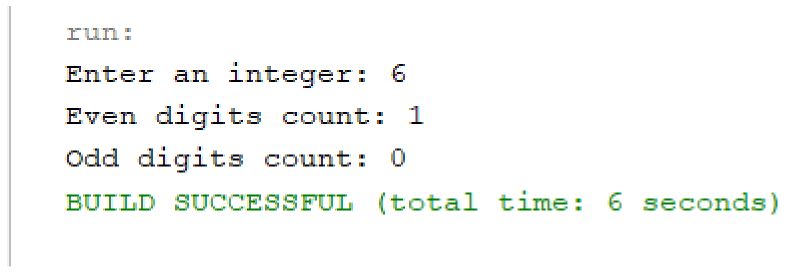
Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

INPUT:



```java
package lab1;
import java.util.Scanner;
public class LAB1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        Integer number = scanner.nextInt();  // Autoboxing
        int evenCount = 0;
        int oddCount = 0;
        while (number != 0) {
            int digit = number % 10;  // Unboxing
            if (digit % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
            number /= 10;  // Remove the last digit
        }
        System.out.println("Even digits count: " + evenCount);
        System.out.println("Odd digits count: " + oddCount);
    }
}
```

OUTPUT:



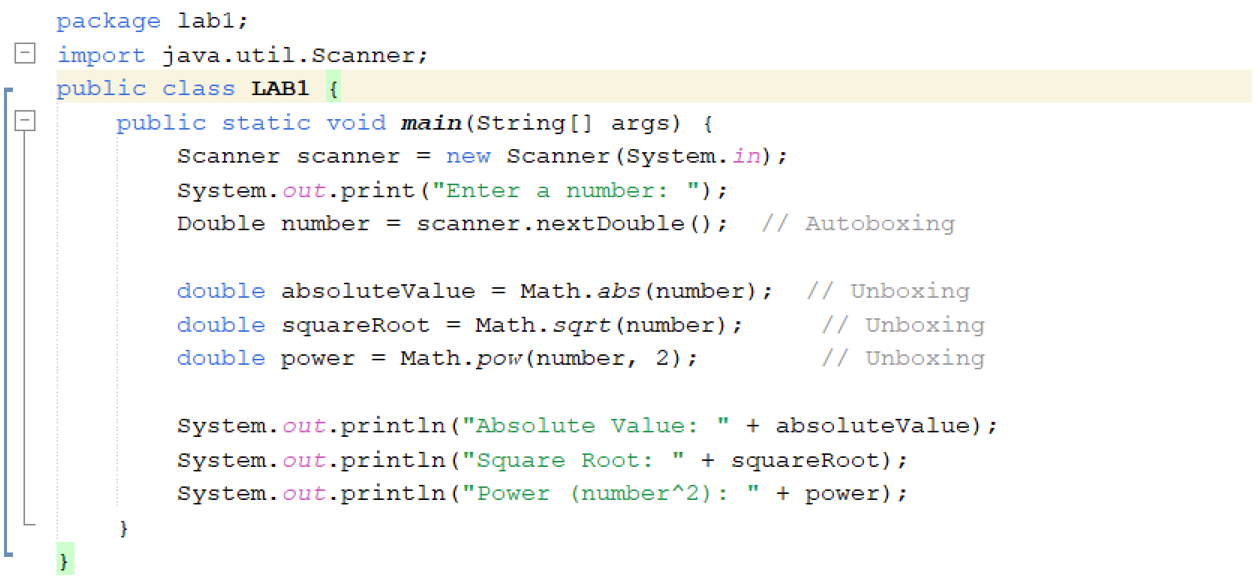
```
run:
Enter an integer: 6
Even digits count: 1
Odd digits count: 0
BUILD SUCCESSFUL (total time: 6 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 3:

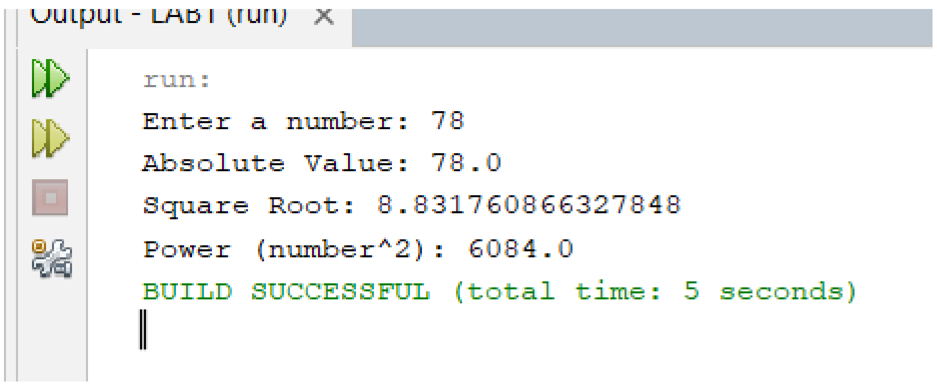
Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes.

INPUT:



```java
package lab1;
import java.util.Scanner;
public class LAB1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        Double number = scanner.nextDouble();  // Autoboxing

        double absoluteValue = Math.abs(number);  // Unboxing
        double squareRoot = Math.sqrt(number);     // Unboxing
        double power = Math.pow(number, 2);        // Unboxing

        System.out.println("Absolute Value: " + absoluteValue);
        System.out.println("Square Root: " + squareRoot);
        System.out.println("Power (number^2): " + power);
    }
}
```

OUTPUT:



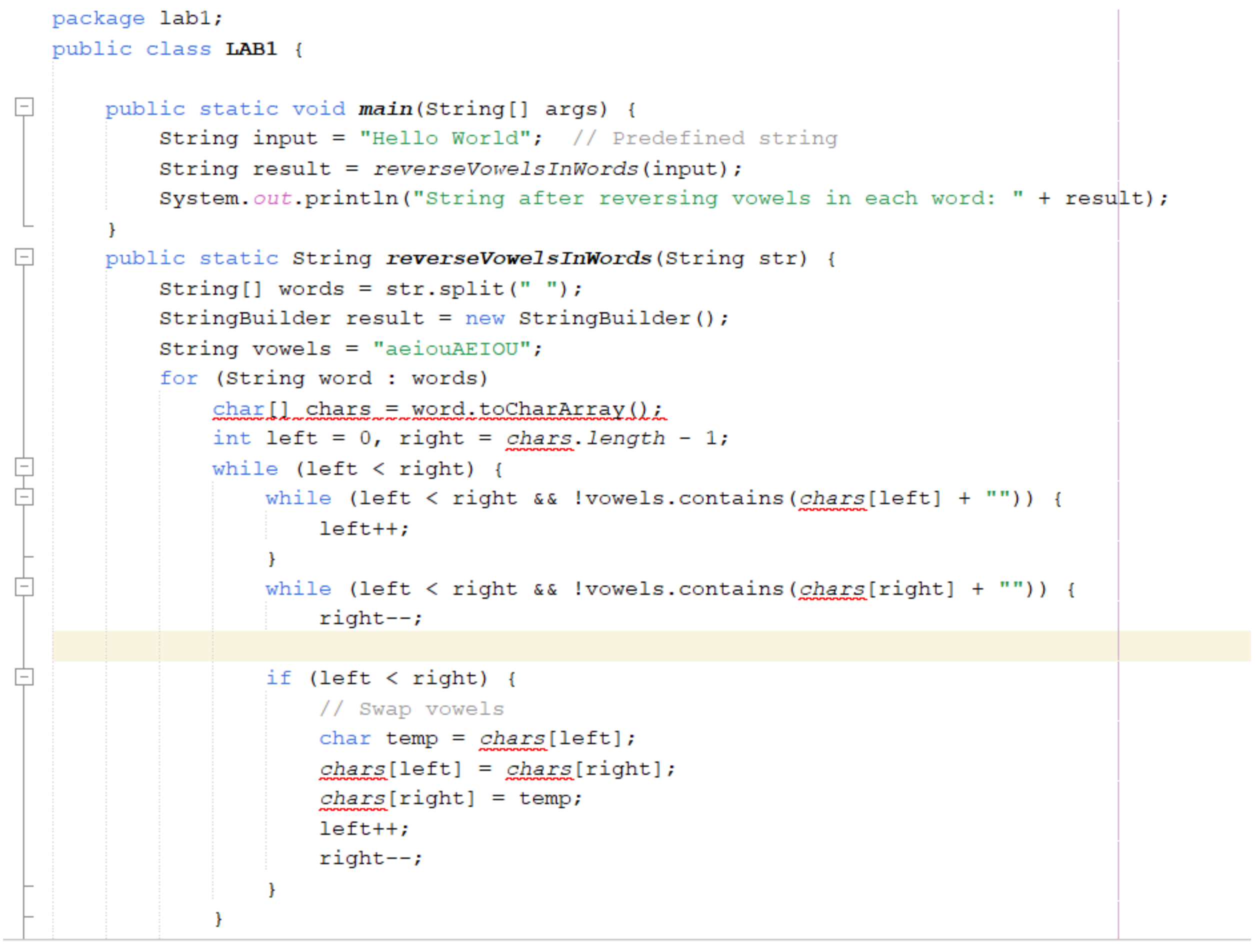
```
run:
Enter a number: 78
Absolute Value: 78.0
Square Root: 8.831760866327848
Power (number^2): 6084.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 4:

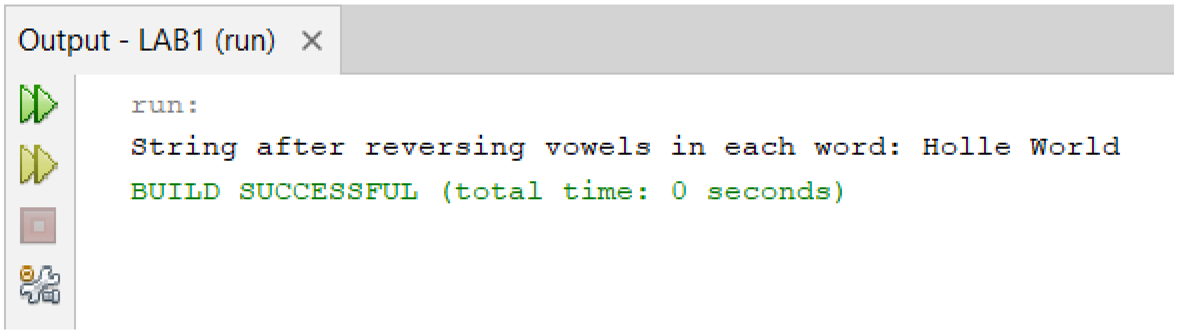
Write a Java program to reverse only the vowels in a string.

INPUT:



```java
package lab1;
public class LAB1 {

    public static void main(String[] args) {
        String input = "Hello World";  // Predefined string
        String result = reverseVowelsInWords(input);
        System.out.println("String after reversing vowels in each word: " + result);
    }
    public static String reverseVowelsInWords(String str) {
        String[] words = str.split(" ");
        StringBuilder result = new StringBuilder();
        String vowels = "aeiouAEIOU";
        for (String word : words)
            char[] chars = word.toCharArray();
            int left = 0, right = chars.length - 1;
            while (left < right) {
                while (left < right && !vowels.contains(chars[left] + "")) {
                    left++;
                }
                while (left < right && !vowels.contains(chars[right] + "")) {
                    right--;

                if (left < right) {
                    // Swap vowels
                    char temp = chars[left];
                    chars[left] = chars[right];
                    chars[right] = temp;
                    left++;
                    right--;
                }
            }
```

OUTPUT:



Output - LAB1 (run)

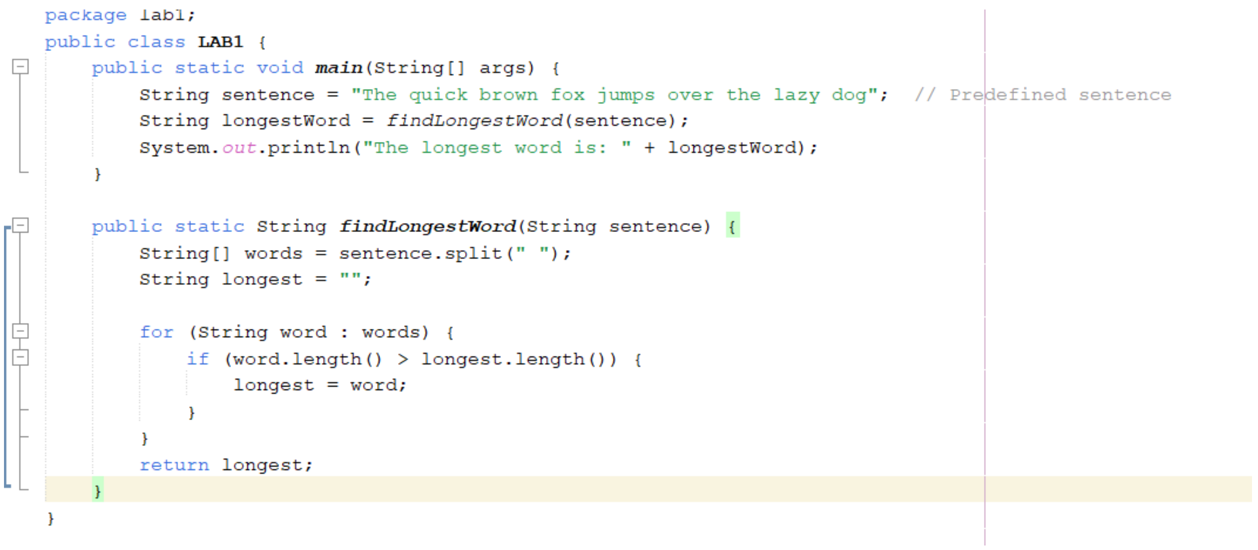
```
run:
String after reversing vowels in each word: Holle World
BUILD SUCCESSFUL (total time: 0 seconds)
```

Name: **Muhammad Moosa Raza** Roll no: **2023F-BSE-375** Section: **E**

TASK NO 5:

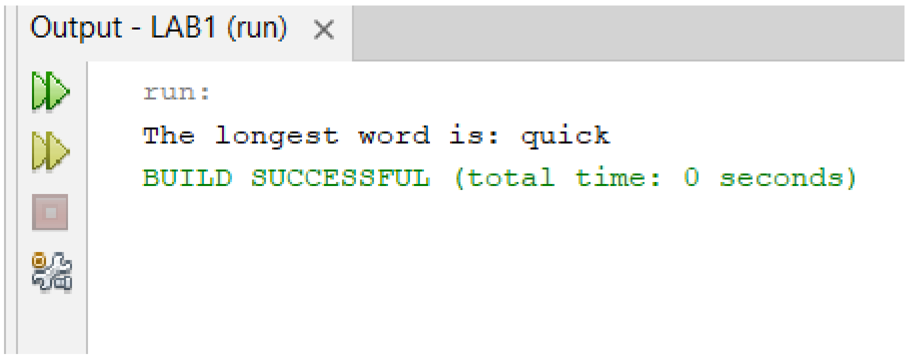
Write a Java program to find the longest word in a sentence

INPUT:



```java
package lab1;
public class LAB1 {
    public static void main(String[] args) {
        String sentence = "The quick brown fox jumps over the lazy dog";  // Predefined sentence
        String longestWord = findLongestWord(sentence);
        System.out.println("The longest word is: " + longestWord);
    }

    public static String findLongestWord(String sentence) {
        String[] words = sentence.split(" ");
        String longest = "";

        for (String word : words) {
            if (word.length() > longest.length()) {
                longest = word;
            }
        }
        return longest;
    }
}
```

OUTPUT:



```
run:
The longest word is: quick
BUILD SUCCESSFUL (total time: 0 seconds)
```