

# Assignment – 7

## Practice Questions : 1

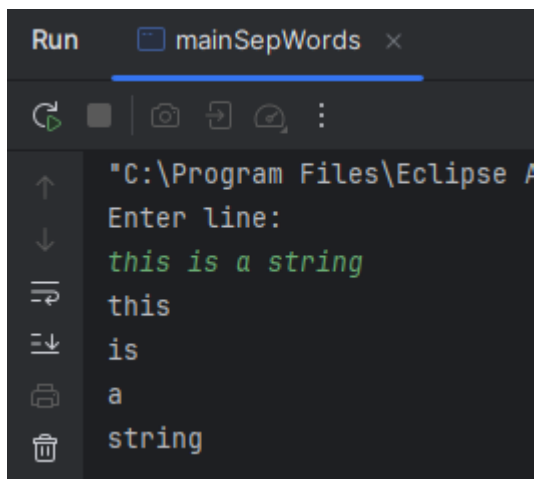
Q1) Accept a sentence from user and print its every word on separate line.

CODE :

```
© SepWords.java × mainSepWords.java
1 package Practice5;
2
3 public class SepWords { 2 usages  ⬆ RevanMidha005
4 @ void sep(String line){ 1 usage  ⬆ RevanMidha005
5     String[] words = line.split(regex: " ", limit: 0);
6     for(String word : words){
7         System.out.println(word);
8     }
9 }
10 }
11 |
```

```
© SepWords.java mainSepWords.java ×
1 package Practice5;
2 import java.util.Scanner;
3
4 ▶ public class mainSepWords { ⬆ RevanMidha005
5 ▶     public static void main(String[] args) { ⬆ RevanMidha005
6         Scanner scn = new Scanner(System.in);
7         System.out.println("Enter line: ");
8         String line = scn.nextLine();
9
10        SepWords obj = new SepWords();
11        obj.sep(line);
12    }
13 }
```

## OUTPUT :



The screenshot shows a Java IDE's console window titled "Run" with a tab for "mainSepWords". The console displays the following output:

```
"C:\Program Files\Eclipse A
Enter line:
this is a string
this
is
a
string
```

The output shows the full path of the Eclipse IDE, followed by the prompt "Enter line:". The user input "this is a string" is shown in green. The program then outputs each word of the input string on a new line: "this", "is", "a", and "string".

**HANDWRITTEN :**

Ans) SepWords.java

```
package Practice5;
```

```
public class SepWords {
```

```
    void sep(String line) {
```

```
        String[] words = line.split(" ", 0);
```

```
        for (String word : words) {
```

```
            System.out.print(word + " ");
```

```
        }
```

```
    }
```

```
}
```

mainSepWords.java

```
package Practice5;
```

```
import java.util.Scanner;
```

```
public class mainSepWords {
```

```
    public static void main (String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter line: ");
```

```
        String line = sc.nextLine();
```

```
        SepWords obj = new SepWords();
```

```
        obj.sep(line);
```

```
    }
```

```
}
```

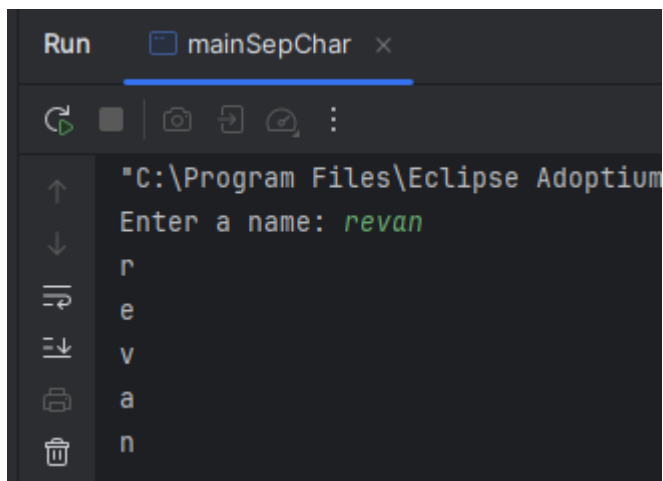
Q2) Accept the name of the user and print all its character on different line.

CODE :

```
SepChar.java × mainSepChar.java
1 package Practice5;
2
3 public class SepChar { 2 usages  ⬆ RevanMidha005
4     void sepChar(String name){ 1 usage  ⬆ RevanMidha005
5         for (int i = 0; i < name.length(); i++){
6             System.out.println(name.charAt(i));
7         }
8     }
9 }
10
```

```
SepChar.java mainSepChar.java ×
1 package Practice5;
2 import java.util.Scanner;
3
4 public class mainSepChar {
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7         System.out.print("Enter a name: ");
8         String name = scn.nextLine();
9
10        SepChar obj = new SepChar();
11        obj.sepChar(name);
12    }
13 }
14
```

## OUTPUT :



The screenshot shows the Eclipse IDE's Run console window. The title bar reads "Run" and "mainSepChar" with a close button. The console output shows the file path "C:\Program Files\Eclipse Adoptium\bin\java.exe" followed by the prompt "Enter a name: revan". The input "revan" is displayed on the next line, with each character on a new line: "r", "e", "v", "a", and "n".

```
"C:\Program Files\Eclipse Adoptium\bin\java.exe"  
Enter a name: revan  
r  
e  
v  
a  
n
```

**HANDWRITTEN :**

my Sepchar.java

```
package Practice5;  
  
public class Sepchar {  
    void sepchar(String name) {  
        for (int i = 0; i < name.length(); i++) {  
            System.out.println(name.charAt(i));  
        }  
    }  
}
```

mainSepchar.java

```
package Practice5;  
import java.util.Scanner;
```

```
public class mainSepchar {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter name :");  
        String name = sc.nextLine();  
  
        Sepchar obj = new Sepchar();  
        obj.sepchar(name);  
    }  
}
```

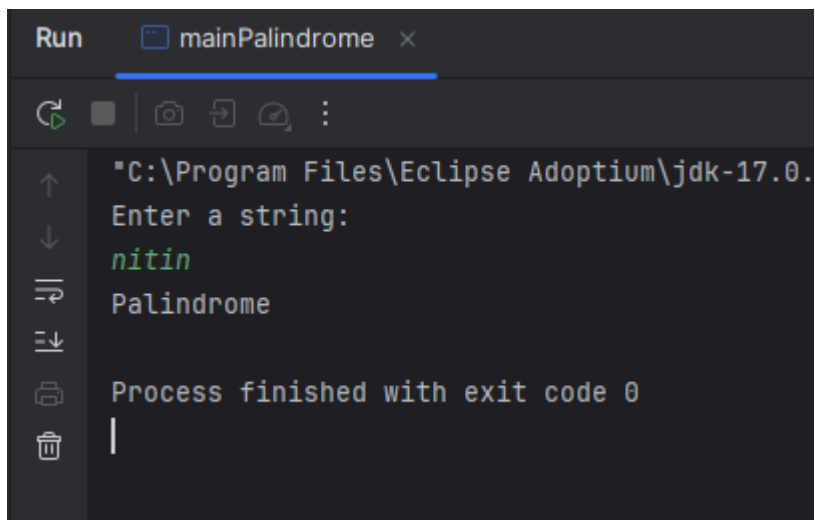
Q3) Accept a string from user and check whether it is palindrome or not.

CODE :

```
© Palindrome.java × mainPalindrome.java
1 package Practice5;
2
3 public class Palindrome { 2 usages  ⬆ RevanMidha005
4     @ boolean isPalin(String str) { 1 usage  ⬆ RevanMidha005
5         for (int i = 0; i < str.length() / 2; i++) {
6             if (str.charAt(i) != str.charAt(str.length() - 1 - i)) {
7                 return false;
8             }
9         }
10        return true;
11    }
12 }
13
```

```
© Palindrome.java mainPalindrome.java ×
1 package Practice5;
2 import java.util.Scanner;
3
4 public class mainPalindrome { ⬆ RevanMidha005
5     public static void main(String[] args) { ⬆ RevanMidha005
6         Scanner scn = new Scanner(System.in);
7         System.out.println("Enter a string: ");
8         String str = scn.nextLine();
9
10        Palindrome obj = new Palindrome();
11
12        boolean check = obj.isPalin(str);
13
14        if (check){
15            System.out.println("Palindrome");
16        }
17        else{
18            System.out.println("Not Palindrome");
19        }
20    }
21 }
22
```

## OUTPUT :



The screenshot shows the Eclipse IDE's Run console window. The title bar reads "Run" and "mainPalindrome x". The console output is as follows:

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.  
Enter a string:  
nitin  
Palindrome  
  
Process finished with exit code 0  
|
```

The output indicates that the program successfully identified "nitin" as a palindrome and finished execution with an exit code of 0.



## HANDWRITTEN :

Ans) palindrome.java

```
package practice5;
```

```
public class palindrome {
```

```
    boolean isPalin(String str) {
```

```
        for (int i = 0; i < str.length() / 2; i++) {
```

```
            if (str.charAt(i) != str.charAt(str.length() - i - 1)) {  
                return false;  
            }
```

```
        }
```

```
        return true;
```

```
    }
```

```
}
```

mainPalin.java

```
package practice5;
```

```
import java.util.Scanner;
```

```
public class mainPalin {
```

```
    public static void main(String[] args) {
```

```
        Scanner scn = new Scanner(System.in);
```

```
        System.out.println("Enter string:");
```

```
        String str = scn.nextLine();
```

```
        palindrome obj = new palindrome();
```

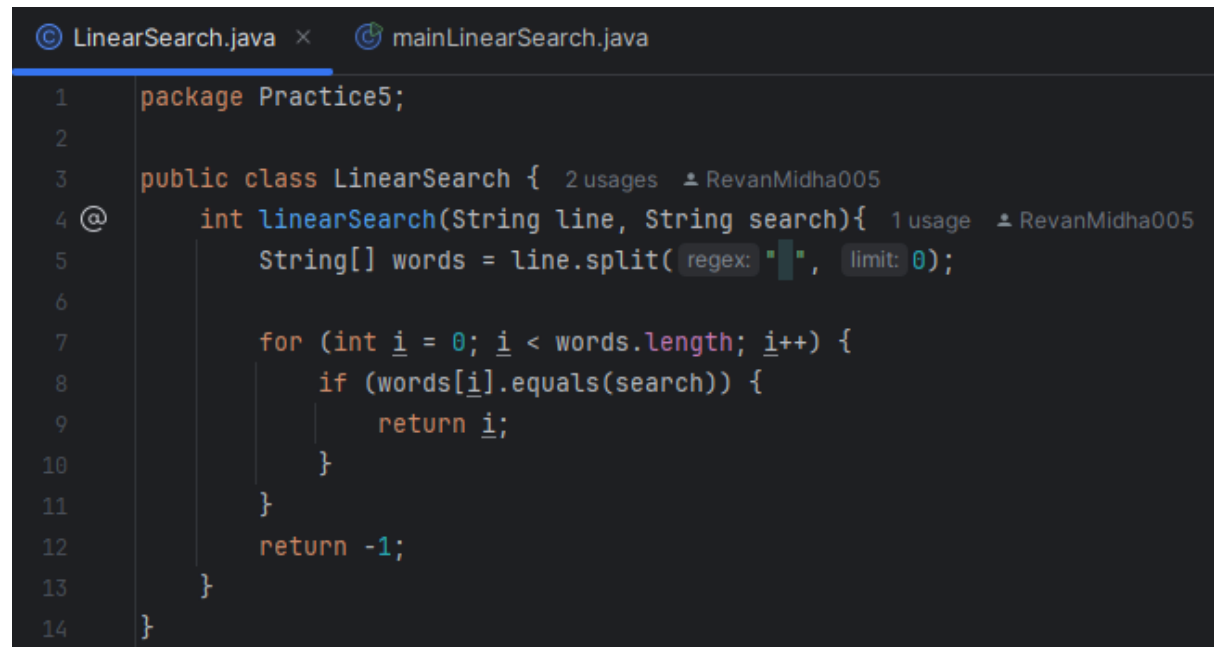
```
        boolean check = obj.isPalin(str);
```

DT.

```
if (check) {  
    System.out.println("palindrome");  
}  
else {  
    System.out.println("Not palindrome");  
}  
}
```

Q4) Accept a line from user and then accept the word to be searched in it. Print the location of it.

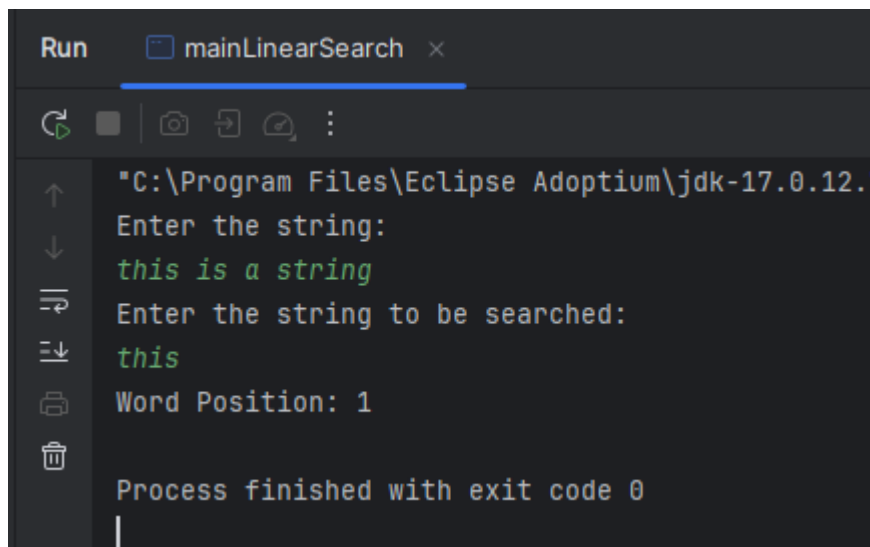
CODE :



```
1 package Practice5;
2
3 public class LinearSearch { 2 usages  ⤴ RevanMidha005
4   @ int linearSearch(String line, String search){ 1 usage  ⤴ RevanMidha005
5       String[] words = line.split(regex: " ", limit: 0);
6
7       for (int i = 0; i < words.length; i++) {
8           if (words[i].equals(search)) {
9               return i;
10          }
11      }
12      return -1;
13  }
14 }
```

```
© LinearSearch.java    mainLinearSearch.java ×
1  package Practice5;
2  import java.util.Scanner;
3
4  ▶ public class mainLinearSearch {  ⤴ RevanMidha005 *
5  ▶      public static void main(String[] args) {  ⤴ RevanMidha005 *
6          Scanner scn = new Scanner(System.in);
7
8          System.out.println("Enter the string: ");
9          String line = scn.nextLine();
10         System.out.println("Enter the string to be searched: ");
11         String search = scn.nextLine();
12
13         LinearSearch obj4 = new LinearSearch();
14         int idx = obj4.linearSearch(line, search);
15
16         if (idx != -1){
17             System.out.println("Word Position: " + (idx + 1));
18         }
19         else{
20             System.out.println("Not found");
21         }
22     }
23 }
24
```

## OUTPUT :



The screenshot shows the Eclipse IDE's Run console window. The title bar indicates the application is 'mainLinearSearch'. The console output is as follows:

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.12.1\bin\java.exe" -Djava.class.path=.\ -Djava.library.path=.\ -Xms128m -Xmx1024m -XX:MaxPermSize=256m -XX:+UseG1GC -XX:+UseStringDeduplication -XX:HeapDumpPath=C:\Program Files\Eclipse Adoptium\jdk-17.0.12.1\bin\hs_err_pid12345.log -jar C:\Program Files\Eclipse Adoptium\jdk-17.0.12.1\bin\java.exe
Enter the string:
this is a string
Enter the string to be searched:
this
Word Position: 1
Process finished with exit code 0
```

## HANDWRITTEN :

Ans) LinearSearch.java

```
package Practice5;
```

```
public class LinearSearch {
```

```
    int linearSearch(String line, String search) {
```

```
        String [] words = line.split(" ", 0);
```

```
        for (int i=0; i < words.length; i++) {
```

```
            if (words[i].equals(search)) {
```

```
                return 1;
```

```
            }
```

```
        }
```

```
        return -1;
```

```
    }
```

```
}
```

mainLinearSearch.java

```
package Practice5;
```

```
import java.util.Scanner;
```

```
public class mainLinearSearch {
```

```
    public static void main (String [] args) {
```

```
        Scanner scn = new Scanner (System.in);
```

```
        System.out.println("Enter string: ");
```

```
        String line = scn.nextLine();
```

```
        System.out.println("Enter string to be searched: ");
```

```
        String search = scn.nextLine();
```

Spiral

DT.

```
LinearSearch obj = new LinearSearch();  
int idx = obj.LinearSearch(line, search);  
  
if (idx != -1) {  
    System.out.println("found at index: " + idx);  
}  
else {  
    System.out.println("Not found");  
}  
}
```

Q5) Accept a String from user and reverse it without using another String variable.

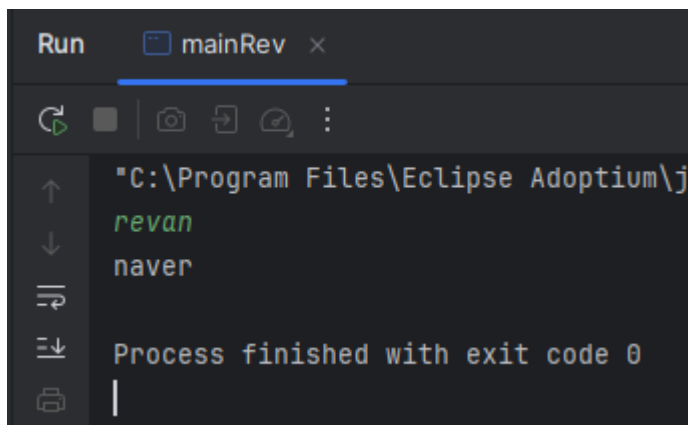
CODE :

```
Reverse.java x mainRev.java
1 package Practice5;
2
3 public class Reverse { 2 usages  ⬆ RevanMidha005
4   @ String rev(String str){ 1 usage  ⬆ RevanMidha005
5       String s = "";
6       for (int i = str.length() - 1; i >= 0; i--){
7           s += str.charAt(i);
8       }
9       return s;
10    }
11 }
12
```

```
Reverse.java mainRev.java x
1 package Practice5;
2 import java.util.Scanner;
3
4 public class mainRev { ⬆ RevanMidha005
5     public static void main(String[] args) { ⬆ RevanMidha005
6         Scanner scn = new Scanner(System.in);
7         String str = scn.nextLine();
8
9         Reverse obj = new Reverse();
10        String revStr = obj.rev(str);
11        System.out.println(revStr);
12    }
13 }
14
```



## OUTPUT :



The screenshot shows the Eclipse IDE's Run console window. The title bar reads "Run" and "mainRev x". The console output is as follows:

```
"C:\Program Files\Eclipse Adoptium\j  
revan  
naver  
  
Process finished with exit code 0  
|
```

The output consists of a file path, the word "revan" in green, the word "naver", a blank line, and a message indicating the process finished with exit code 0. A cursor is visible at the end of the last line.

## HANDWRITTEN :

```
Ans) Reverse.java
package Practices;
public class Reverse {
    String new(String str) {
        String s = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            s += str.charAt(i);
        }
        return s;
    }
}
```

main Rev.java

```
package Practices;
import java.util.Scanner;

public class main Rev {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        Reverse obj = new Reverse();
        String newStr = obj.new(str);
        System.out.println(newStr);
    }
}
```

Q6) Write a method to take input in two matrix from user and display the product of matrix

CODE :

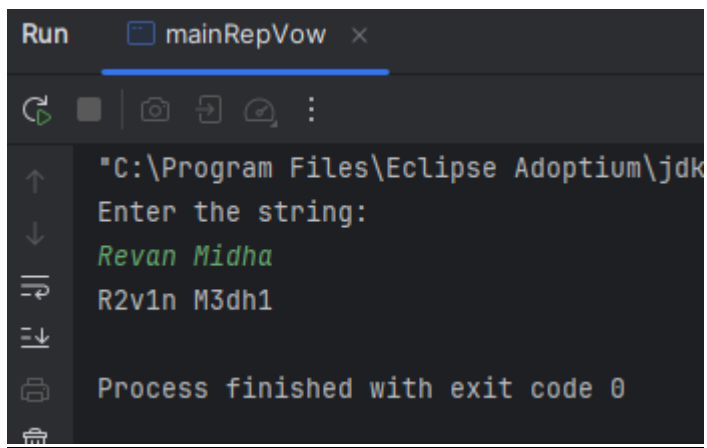
```
ReplaceVow.java × mainRepVow.java
1 package Practice5;
2
3 public class ReplaceVow { 2 usages RevanMidha005
4     @ String rep(String line){ 1 usage RevanMidha005
5         String s = "";
6         for (char i : line.toCharArray()){
7             char x = Character.toLowerCase(i);
8
9             if (x == 'a'){
10                 s += "1";
11             }
12             else if (x == 'e'){
13                 s += "2";
14             }
15             else if (x == 'i'){
16                 s += "3";
17             }
18             else if (x == 'o'){
19                 s += "4";
20             }
21             else if (x == 'u'){
22                 s += "5";
23             }
24             else{
25                 s += i;
26             }
27         }
28         return s;
29     }
```

© ReplaceVow.java

© mainRepVow.java ×

```
1 package Practice5;
2 import java.util.Scanner;
3
4 public class mainRepVow {  ± RevanMidha005
5     public static void main(String[] args) {  ± RevanMidha005
6         Scanner scn = new Scanner(System.in);
7         System.out.println("Enter the string: ");
8         String str = scn.nextLine();
9
10        ReplaceVow obj = new ReplaceVow();
11        String replacedStr = obj.rep(str);
12        System.out.println(replacedStr);
13    }
14 }
15
```

## OUTPUT :



The screenshot shows the 'Run' console window in an Eclipse IDE. The window title is 'Run' with a sub-tab 'mainRepVow'. The console output is as follows:

```
"C:\Program Files\Eclipse Adoptium\jdk
Enter the string:
Revan Midha
R2v1n M3dh1
Process finished with exit code 0
```

On the left side of the console, there is a vertical toolbar with icons for: running, stopping, debugging, and other actions.

# HANDWRITTEN :

main) ReplaceVow.java

~~package~~ package Practice5;

public class ReplaceVow {

String replace(String line) {

String s = "";

for (char i: line.toCharArray()) {  
char x = Character.toLowerCase(i);

if (x == 'a') {  
s += "1";

}  
else if (x == 'e') { s += "2"; }

else if (x == 'i') { s += "3"; }

else if (x == 'o') { s += "4"; }

else if (x == 'u') { s += "5"; }

else {  
s += i;

}

}  
return s;

}

}

mainRepVow.java

package practice5;  
import java.util.Scanner;

public class mainRepVow {

public static void main (String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter str:");

String str = sc.nextLine();

ReplaceVow obj = new ReplaceVow();

String replacedStr = obj.rep(str);

System.out.println(replacedStr);

3  
}