Program 1:

Playing with String - I

Given a string array and non negative integer (n) apply the following rules.

1. Pick nth character from each String element in the String array and form a new String.
2. If nth character not available in a particular String in the array consider $ as the character.
3. Return the newly formed string.

Include a class UserMainCode with a static method formString which accepts the string and integer. The return type is the string formed based on rules.

Create a Class Main which would be used to accept the string and integer and call the static method present in UserMainCode.

```
import java.util.Scanner;

public class UserMainCode {
        static String formString() {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the size of the array:");
                int size = sc.nextInt();
                String str[] = new String[size];
                System.out.println("Please Enter the strings");
                for (int i = 0; i < size; i++) {
                        str[i] = sc.next();
                }
                System.out.println("Please enter the non negative integer:");
                int n = sc.nextInt() - 1;
                String result = "";
                for (int j = 0; j < str.length; j++) {
                        if (str[j].length() > n)

                                result = result + str[j].charAt(n);
                        else
                                result = result + "$";

                }

                return result;
        }
}
```

```
public class Main {

        public static void main(String[] args) {
                UserMainCode usermaincode = new UserMainCode();
                System.out.println(UserMainCode.formString());

        }

}
```

program 2:

Reverse SubString
Given a string, startIndex and length, write a program to extract the substring from right to left. Assume the last character has index 0.
Include a class UserMainCode with a static method “reverseSubstring” that accepts 3 arguments and returns a string. The 1st argument corresponds to the string, the second argument corresponds to the startIndex and the third argument corresponds to the length.
Create a class Main which would get a String and 2 integers as input and call the static method reverseSubstring present in the UserMainCode.

```
public class ReverseSubstring {

        static String reverseSubstring(String str, int startindex, int endindex) {
                StringBuffer sb = new StringBuffer(str);
                sb.reverse();
                String str1 = sb.substring(startindex, endindex);
                return str1;

        }
}

import java.util.Scanner;

public class TestReverseString {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.out.println("Enter the string ,starting index and ending index:");

                Scanner sc=new Scanner(System.in);
        String str=sc.next();
        int startindex=sc.nextInt();
        int endindex=sc.nextInt();
        ReverseSubstring reversesubstring=new ReverseSubstring();

System.out.println(reversesubstring.reverseSubstring(str,startindex,endindex));




    }

}
```

program 3:

Fetching Middle Characters from String
Write a program to read a string of even length and to fetch two middle most characters from the input string and return it as string output.
Include a class UserMainCode with a static method getMiddleChars which accepts a string of even length as input . The return type is a string which should be the middle characters of the string.
Create a class Main which would get the input as a string and call the static method getMiddleChars present in the UserMainCode.

```
public class MiddleCharacter {
    public static String getMiddleChars(String str) {
        StringBuffer sb = new StringBuffer();
        if (str.length() % 2 == 0) {
            sb.append(str.substring((str.length() / 2) - 1, (str.length() / 2) +
1));
        }
        return sb.toString();
    }
}

import java.util.Scanner;

public class TestMiddleChars {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter the string:");
        Scanner sc = new Scanner(System.in);
```

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

            String s1 = MiddleCharacter.getMiddleChars(s);
            System.out.println(s1);

        }

}
```

program 4:

String processing – Long + Short + Long
Obtain two strings S1,S2 from user as input. Your program should form a string of "long+short+long", with the shorter string inside of the longer String.
Include a class UserMainCode with a static method getCombo which accepts two string variables. The return type is the string.
Create a Class Main which would be used to accept two Input strings and call the static method present in UserMainCode.
Input and Output Format:
Input consists of two strings with maximum size of 100 characters.
Output consists of an string.

```
public class StringCombo {

        static String getCombo(String str1, String str2) {
                StringBuffer sb = new StringBuffer();
                int len1 = str1.length();
                int len2 = str2.length();

                if (len1 > len2)
                        System.out.println(str1 + str2 + str1);

                else if (len2 > len1)

                        System.out.println(str2 + str1 + str2);
                else
                        System.out.println(str1 + " " + str2);
                return sb.toString();

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

```
public class TestStringCombo {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter two strings:");
                String str1 = sc.next();
                String str2 = sc.next();
                StringCombo stringcombo = new StringCombo();
                stringcombo.getCombo(str1, str2);

        }

}
```

program 5:

Strings Processing - Replication
Write a program to read a string and also a number N. Return the replica of original string for n given time.
Include a class UserMainCode with a static method repeatString which accepts the the string and the number n. The return type is the string based on the problem statement.
Create a Class Main which would be used to accept the string and integer and call the static method present in UserMainCode.

```
public class Replica {
        static String repeatString(String str, int n) {
                StringBuffer sb = new StringBuffer();
                for (int i = 1; i <= n; i++)
                        sb.append(str);

                return sb.toString();

        }
}
```

```
import java.util.Scanner;

public class TestReplica {

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                // TODO Auto-generated method stub
                System.out.println("Enter a String and the number of times :");
                String str = sc.next();
                int n = sc.nextInt();
                Replica replica = new Replica();
                replica.repeatString(str, n);
                System.out.println(replica.repeatString(str, n));

        }
}
```

program 6:

Flush Characters
Write a program to read a string from the user and remove all the alphabets and spaces from the String, and only store special characters and digit in the output String. Print the output string.
Include a class UserMainCode with a static method getSpecialChar which accepts a string. The return type (String) should return the character removed string.
Create a Class Main which would be used to accept a string and call the static method present in UserMainCode.

```
public class FlushCharacter {
        public static String getSpecialChars(String s1) {
                return s1.replaceAll("[A-Za-z]", "");

        }
}

import java.util.Scanner;

public class TestFlushCharacters {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter mixture of alphabets ,specialcharcters and
integers:");
                String s1 = sc.nextLine();
                FlushCharacter flushcharacter = new FlushCharacter();
                System.out.println(flushcharacter.getSpecialChars(s1));
        }

}
```

program 7:

Negative String
Given a string input, write a program to replace every appearance of the word "is" by "is not".
If the word "is" is immediately preceeded or followed by a letter no change should be made to the string .
Include a class UserMainCode with a static method “negativeString” that accepts a String arguement and returns a String.
Create a class Main which would get a String as input and call the static method negativeString present in the UserMainCode.

```
public class ReplaceIs {
        public static String negativeString(String str) {
                return str.replaceAll(" is", " is not");

        }
}

import java.util.Scanner;

public class TestReplaceIs {

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a sentence");
                String str = sc.nextLine();
                // TODO Auto-generated method stub
                ReplaceIs replaceis = new ReplaceIs();
                System.out.println(replaceis.negativeString(str));

        }

}
```

program 8:

Name Shrinking
Write a program that accepts a string as input and converts the first two names into dot-separated initials and printa the output.
Input string format is 'fn mn ln'. Output string format is 'ln [mn's 1st character].[fn's 1st character]'
Include a class UserMainCode with a static method getFormatedString which accepts a string. The return type (String) should return the shrinked

name.
Create a Class Main which would be used to accept Input String and call the static method present in UserMainCode.

```
import java.util.StringTokenizer;

public class Initial {
        static void getFormatedString(String str) {
                StringBuffer sb = new StringBuffer();
                StringTokenizer st = new StringTokenizer(str, " ");
                String s1 = st.nextToken();
                String s2 = st.nextToken();
                String s3 = st.nextToken();
                sb.append(s3);
                sb.append(" ");
                sb.append(s2.substring(0, 1));
                sb.append(".");
                sb.append(s1.substring(0, 1));
                System.out.println(sb);
        }

}
```

```
import java.util.StringTokenizer;

public class Initial {
        static void getFormatedString(String str) {
                StringBuffer sb = new StringBuffer();
                StringTokenizer st = new StringTokenizer(str, " ");
                String s1 = st.nextToken();
                String s2 = st.nextToken();
                String s3 = st.nextToken();
                sb.append(s3);
                sb.append(" ");
                sb.append(s2.substring(0, 1));
                sb.append(".");
                sb.append(s1.substring(0, 1));
                System.out.println(sb);
        }

}
```

Program 9:

Start Case

Write a program to read a sentence in string variable and convert the first letter of each word to capital case. Print the final string.

Note: - Only the first letter in each word should be in capital case in final string.

Include a class UserMainCode with a static method printCapitalized which accepts a string. The return type (String) should return the capitalized string.

Create a Class Main which would be used to accept a string and call the static method present in UserMainCode.

```
import java.util.StringTokenizer;

public class Captilize {
        public static String printCapitilized(String str) {
                StringBuffer sb = new StringBuffer();
                StringTokenizer st = new StringTokenizer(str, " ");
                while (st.hasMoreTokens()) {
                        String s1 = st.nextToken();
                        String s2 = s1.substring(0, 1);
                        String s3 = s1.substring(1, s1.length());
                        sb.append(s2.toUpperCase()).append(s3).append(" ");
                }
                return sb.toString();

        }

}
```

```
import java.util.StringTokenizer;

public class Captilize {
        public static String printCapitilized(String str) {
                StringBuffer sb = new StringBuffer();
                StringTokenizer st = new StringTokenizer(str, " ");
                while (st.hasMoreTokens()) {
                        String s1 = st.nextToken();
                        String s2 = s1.substring(0, 1);
                        String s3 = s1.substring(1, s1.length());
                        sb.append(s2.toUpperCase()).append(s3).append(" ");
                }
                return sb.toString();

        }
}
```

program 10:

Occurance Count
Write a program to read a string that contains a sentence and read a word. Check the number of occurances of that word in the sentence. Include a class UserMainCode with a static method countWords which accepts the two strings. The return type is the integer giving the count.
Note: The check is case-sensitive.
Create a Class Main which would be used to accept the two strings and call the static method present in UserMainCode.

```
public class Occurence {

	public static int countWords(String s1, String s2) {
		int count = 0;
		String arr[] = s1.split(" ");
		int l1 = arr.length;
		for (int i = 0; i < l1; i++) {
			if (s2.equals(arr[i]))
				count++;

		}
		return count++;

	}
}

import java.util.Scanner;

public class TestOccurence {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the senstence:");
		String s1 = sc.nextLine();
		System.out.println("Enter the word:");
		String s2 = sc.nextLine();
		Occurence occurence = new Occurence();
		System.out.println(occurence.countWords(s1, s2));
```

```
    }

}
```

Program 11:

String Processing - III
Write a program to read a string where all the lowercase 'x' chars have been moved to the end of the string.
Include a class UserMainCode with a static method moveX which accepts the string. The return type is the modified string.
Create a Class Main which would be used to accept the string and call the static method present in UserMainCode.

```
public class MoveXtoLast {
    static String moveX(String str) {

        String s1 = str.replaceAll("[x]", "");
        // System.out.println(s1);
        String s2 = str.replaceAll("[^x]", "");
        // System.out.println(s2);

        return s1 + s2;

    }
}

import java.util.Scanner;

public class TestMoveX {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String:");
        String str = sc.next();
        MoveXtoLast moveXtolast = new MoveXtoLast();
        System.out.println(moveXtolast.moveX(str));
        // TODO Auto-generated method stub

    }

}
```