

Q2) Write a Java program that uses regular expressions to make a modified copy of a text file that contains, among other things, some 10-digit phone numbers that begin with +966. The modified copy will contain the same text but in every phone number +966 must be replaced with 00966. Test your program well to make sure that it works correctly with all phone numbers regardless of their position in the file (beginning, middle, or end of a line). Also, if the text file contains a + symbol as in 2+3=5 that has nothing to do with phone numbers, it should be copied as it is without any modifications. Use a file that contains the following text to test your program. "My phone number in SA is +9665930234. This phone number will appear in the output file as 009665930234. Even if I write it at the end of the line as in this line +9665930234 Or at the beginning of the line as in the following line +9665930234. It will appear as 009665930234 every time in the output file. However, the + in 2+3=5 will appear as it is without any modification"

Solution:

```
import java.io.*;
import java.nio.file.*;
import java.util.regex.*;

public class PhoneNumberModifier {
    public static void main(String[] args) throws IOException {
        String outputFileName = "output.txt";

        // تعريف نص المدخل مباشرة في البرنامج
        String input = "My phone number in SA is +9665930234. This phone number " +
            "will appear in the output file as 009665930234. Even if I write it " +
            "at the end of the line as in this line +9665930234 ";

        // استخدام تعبير نمطي للبحث عن أرقام الهواتف التي تبدأ ب +966
        Pattern pattern = Pattern.compile("\\+966\\d{7}");

        // استبدال +966 ب 00966 لكل رقم هاتف
        Matcher matcher = pattern.matcher(input);
        String modifiedContent = matcher.replaceAll(match -> "00966" + match.group().substring(4));

        // كتابة النص المعدل إلى ملف جديد
        Files.write(Paths.get(outputFileName), modifiedContent.getBytes());

        System.out.println("The file has been modified and saved as " + outputFileName);
    }
}
```

Q1) Write a Java program that uses regular expressions to count the number of academic courses in a text file that belong to each of the Computer Science department, Information Systems department, and Engineering department. (You have to enter some academic courses codes in a text file, and then display the number of courses in each department) For example: CSC 340 CEN 341 CSC 113 IS 230 Computer Science department = 2 courses Information Systems department = 1 course Engineering department = 1 course

Solution:

```
import java.util.regex.*;

public class CourseCounter {
    public static void main(String[] args) {

        // قراءة البيانات من ملف نصي
        String content = "CSC 340 CEN 341 CSC 113 IS 230";

        // إنشاء تعبيرات نمطية لأقسام الحاسوب ونظم المعلومات والهندسة
        Pattern patternCS = Pattern.compile("CSC \\d+");
        Pattern patternIS = Pattern.compile("IS \\d+");
        Pattern patternEN = Pattern.compile("CEN \\d+");

        // استخدام المطابقة لإيجاد عدد المقررات لكل قسم
        Matcher matcherCS = patternCS.matcher(content);
        Matcher matcherIS = patternIS.matcher(content);
        Matcher matcherEN = patternEN.matcher(content);

        int countCS = 0, countIS = 0, countEN = 0;

        while (matcherCS.find()) {
            countCS++;
        }

        while (matcherIS.find()) {
            countIS++;
        }

        while (matcherEN.find()) {
            countEN++;
        }

        // طباعة النتائج
        System.out.println("Computer Science department = " + countCS + " courses");
        System.out.println("Information Systems department = " + countIS + " courses");
        System.out.println("Engineering department = " + countEN + " courses");
    }
}
```