**package** BankMid;

**public class** StringMatchingInCharValue {

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

String text = **"abcdefgh"**;

String pattern = **"bc"**;

**int** index =0;

**int** lengthOfPattern = pattern.length();

**int** hasCodeForPattern =0;

**int** getASCII = 0;

**char**[] charPattern = pattern.toCharArray();

**for** (**int** i=0;i<charPattern.**length**;i++)

{

getASCII = charPattern[i];

hasCodeForPattern = hasCodeForPattern + getASCII;

}

*printMatchOrNot*(text,hasCodeForPattern,pattern,lengthOfPattern,index);

}

**public static void** printMatchOrNot(String text, **int** hashForPattern,String pattern,**int** lengthOfPattern,**int** index)

{

**int** charCount = 0;

String storeTextChar = **""**;

**int** count = 0;

String afterDeletingFirstChar =**""**;

**int** l = 1;

**int** length = 0;

**int** startingIndex = 0;

**if** (text.length()<pattern.length())

{

System.***out***.println(**"Pattern is not match: "**);

}**else** {

**while** (lengthOfPattern > 0) {

storeTextChar = storeTextChar + text.charAt(charCount);

charCount = charCount + 1;

lengthOfPattern = lengthOfPattern - 1;

}

**int** hashForText = *HashCodeGenerate*(storeTextChar);

**if** (hashForText == hashForPattern) {

**for** (**int** i = 0; i < pattern.length(); i++) {

**if** (text.charAt(i) == pattern.charAt(i)) {

count = count + 1;

startingIndex = index;

} **else**

**break**;

}

} **else** {

**char**[] ch = text.toCharArray();

**for** (**int** i = l; i < ch.**length**; i++) {

afterDeletingFirstChar = afterDeletingFirstChar + ch[i];

}

length = pattern.length();

}

**if** (count == pattern.length()) {

System.***out***.println(**"Match the pattern: "** + **"'"** + pattern + **"'"** + **", at the starting index is: "** + startingIndex);

} **else** {

*printMatchOrNot*(afterDeletingFirstChar, hashForPattern, pattern, length, index + 1);

}

}

}

**public static int** HashCodeGenerate(String textString)

{

**int** getASCII ;

**int** hashFor = 0;

**char**[] ch = textString.toCharArray();

**for** (**int** i=0;i<ch.**length**;i++)

{

getASCII = ch[i];

hashFor = hashFor + getASCII;

}

**return** hashFor;

}

}