|  |
| --- |
| /\*\*  \*\* Java Program to implement Boyer Moore Algorithm  \*\*/  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.io.IOException;  /\*\* Class BoyerMoore \*\*/  public class BoyerMoore  {  /\*\* function findPattern \*\*/  public void findPattern(String t, String p)  {  char[] text = t.toCharArray();  char[] pattern = p.toCharArray();  int pos = indexOf(text, pattern);  if (pos == -1)  System.out.println("\nNo Match\n");  else  System.out.println("Pattern found at position : "+ pos);  }  /\*\* Function to calculate index of pattern substring \*\*/  public int indexOf(char[] text, char[] pattern)  {  if (pattern.length == 0)  return 0;  int charTable[] = makeCharTable(pattern);  int offsetTable[] = makeOffsetTable(pattern);  for (int i = pattern.length - 1, j; i < text.length;)  {  for (j = pattern.length - 1; pattern[j] == text[i]; --i, --j)  if (j == 0)  return i;  i += Math.max(offsetTable[pattern.length - 1 - j], charTable[text[i]]);  }  return -1;  }  private int[] makeCharTable(char[] pattern)  {  final int ALPHABET\_SIZE = 256;  int[] table = new int[ALPHABET\_SIZE];  for (int i = 0; i < table.length; ++i)  table[i] = pattern.length;  for (int i = 0; i < pattern.length - 1; ++i)  table[pattern[i]] = pattern.length - 1 - i;  return table;  }  private static int[] makeOffsetTable(char[] pattern)  {  int[] table = new int[pattern.length];  int lastPrefixPosition = pattern.length;  for (int i = pattern.length - 1; i >= 0; --i)  {  if (isPrefix(pattern, i + 1))  lastPrefixPosition = i + 1;  table[pattern.length - 1 - i] = lastPrefixPosition - i + pattern.length - 1;  }  for (int i = 0; i < pattern.length - 1; ++i)  {  int slen = suffixLength(pattern, i);  table[slen] = pattern.length - 1 - i + slen;  }  return table;  }  private static boolean isPrefix(char[] pattern, int p)  {  for (int i = p, j = 0; i < pattern.length; ++i, ++j)  if (pattern[i] != pattern[j])  return false;  return true;  }  private static int suffixLength(char[] pattern, int p)  {  int len = 0;  for (int i = p, j = pattern.length - 1; i >= 0 && pattern[i] == pattern[j]; --i, --j)  len += 1;  return len;  }  /\*\* Main Function \*\*/  public static void main(String[] args) throws IOException  {  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  System.out.println("Boyer Moore Algorithm Test\n");  System.out.println("\nEnter Text\n");  String text = br.readLine();  System.out.println("\nEnter Pattern\n");  String pattern = br.readLine();  BoyerMoore bm = new BoyerMoore();  bm.findPattern(text, pattern);  }  } |

Output :

