Code the Knuth-Morris-Pratt (KMP) algorithm in C# for pattern searching which pre-processes the pattern to reduce the number of comparisons. Explain how this pre-processing improves the search time compared to the naive approach.

```
public class KMPAlgorithmDemo {

    private final int R;
    private int[][] dfa;
    private char[] pattern;

    public KMPAlgorithmDemo(String pattern) {
        this.R=256;
        this.pattern=pattern.toCharArray();
        int m=pattern.length();
        dfa=new int[R][m];

        dfa[pattern.charAt(0)][0]=1;
        for(int X=0,j=1;j<m;j++) {
            for(int c=0;c<R;c++) {
                dfa[c][j]=dfa[c][X];
            }//for
            dfa[pattern.charAt(j)][j]=j+1;
            X=dfa[pattern.charAt(j)][X];
        }//for
    }

    public void search(String txt) {
        int n=txt.length();
        int m=pattern.length;
        int i,j;
        for(i=0,j=0;i<n&&j<m;i++) {
            j=dfa[txt.charAt(i)][j];
        }//for
        if(j==m) {
            System.out.println("Pattern found at index : "+(i-m));
        }
        else {
            System.out.println("Pattern not found");
        }

    }

    public static void main(String[] args) {
        String txt="ABABDABACDABABCABAB";
        String pat="ABABCABAB";
        KMPAlgorithmDemo kmp=new KMPAlgorithmDemo(pat);
        kmp.search(txt);
    }

}
```