**package** lp3;

**public** **class** NaiveStringMatching {

**public** **static** **void** naiveMatch(String text, String pattern) {

**int** n = text.length();

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

**for** (**int** i = 0; i <= n - m; i++) {

**int** j;

**for** (j = 0; j < m; j++) {

**if** (text.charAt(i + j) != pattern.charAt(j))

**break**;

}

**if** (j == m) {

System.***out***.println("Pattern found at index " + i);

}

}

}

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

String text = "ABABABABCABABABAABABABAB";

String pattern = "ABAB";

System.***out***.println("Naive String Matching:");

*naiveMatch*(text, pattern);

}

}

**package** lp3;

**public** **class** RabinKarpStringMatching {

**private** **static** **final** **int** ***BASE*** = 256; // Base for hash calculation

**private** **static** **final** **int** ***MOD*** = 101; // A prime number for hash calculation

**public** **static** **void** rabinKarpMatch(String text, String pattern) {

**int** n = text.length();

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

**int** patternHash = *calculateHash*(pattern, m);

**int** textHash = *calculateHash*(text, m);

**for** (**int** i = 0; i <= n - m; i++) {

**if** (patternHash == textHash) {

**int** j;

**for** (j = 0; j < m; j++) {

**if** (text.charAt(i + j) != pattern.charAt(j))

**break**;

}

**if** (j == m) {

System.***out***.println("Pattern found at index " + i);

}

}

**if** (i < n - m) {

textHash = *recalculateHash*(textHash, m, text.charAt(i), text.charAt(i + m));

}

}

}

**private** **static** **int** calculateHash(String str, **int** length) {

**int** hash = 0;

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

hash = (hash \* ***BASE*** + str.charAt(i)) % ***MOD***;

}

**return** hash;

}

**private** **static** **int** recalculateHash(**int** oldHash, **int** length, **char** oldChar, **char** newChar) {

**int** newHash = (oldHash - oldChar \* (**int**) Math.*pow*(***BASE***, length - 1)) % ***MOD***;

newHash = (newHash \* ***BASE*** + newChar) % ***MOD***;

**if** (newHash < 0) {

newHash += ***MOD***;

}

**return** newHash;

}

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

String text = "ABABABABCABABABAABABABAB";

String pattern = "ABAB";

System.***out***.println("Rabin-Karp String Matching:");

*rabinKarpMatch*(text, pattern);

}

}

**package** lp3;

**public** **class** StringMatchingComparison {

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

String text = "ABABABABCABABABAABABABAB";

String pattern = "ABAB";

System.***out***.println("Naive String Matching:");

NaiveStringMatching.*naiveMatch*(text, pattern);

System.***out***.println("\nRabin-Karp String Matching:");

RabinKarpStringMatching.*rabinKarpMatch*(text, pattern);

}

}