Experiment No. 12
Naïve String matching
Date of Performance:
Date of Submission:

Experiment No. 12

Title: Naïve String matching

Aim: To study and implement Naïve string matching Algorithm

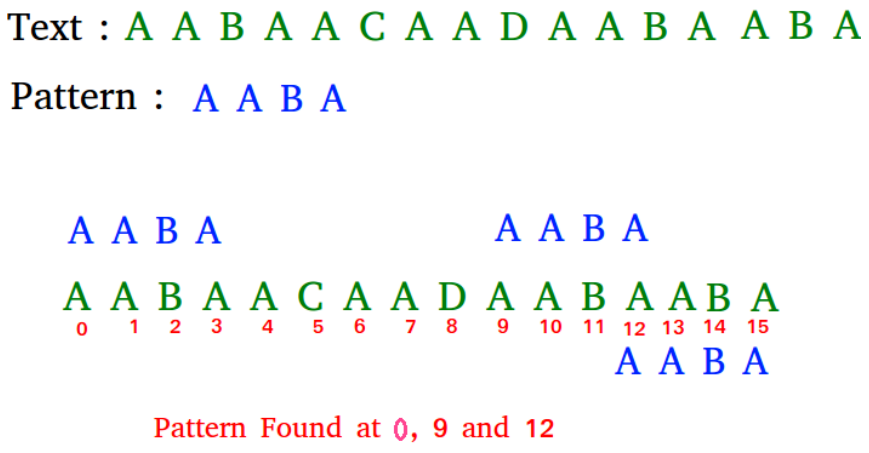
Objective: To introduce String matching methods

Theory:

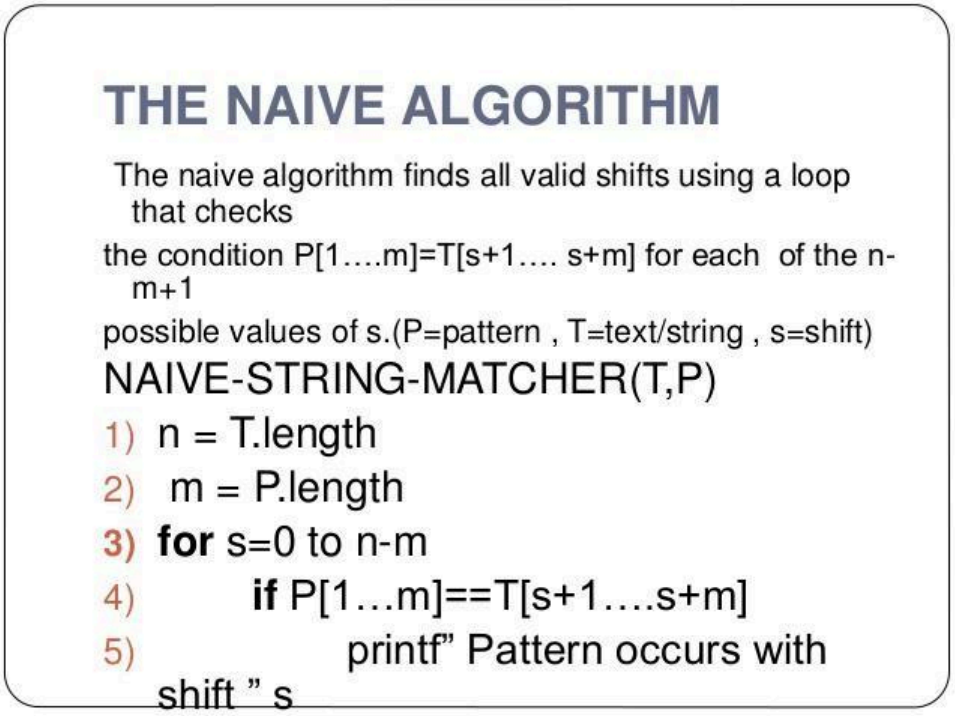
The naïve approach tests all the possible placement of Pattern P [1.....m] relative to text T [1.....n]. We try shift s = 0, 1.....n-m, successively and for each shift s. Compare T [s+1.....s+m] to P [1.....m].

The naïve algorithm finds all valid shifts using a loop that checks the condition P [1.....m] = T [s+1.....s+m] for each of the n - m +1 possible value of s.

Example:



Algorithm:



Implementation:

```
#include <stdio.h>

#include <string.h>


void naiveStringMatch(char text[], char pattern[]) {
```

```
    int m = strlen(pattern);
    int n = strlen(text);

    for (int i = 0; i <= n - m; i++) {
        int j;
        for (j = 0; j < m; j++) {
            if (text[i + j] != pattern[j])
                break;
        }

        if (j == m) {
            printf("Pattern found at index %d\n", i);
        }
    }
}

int main() {
    char text[] = "ABABDABACDABABCABAB";
    char pattern[] = "ABABCABAB";

    printf("Text: %s\n", text);
    printf("Pattern: %s\n", pattern);
```

```
    printf("Occurrences:\n");

    naiveStringMatch(text, pattern);


    return 0;
}
```

Conclusion: Naïve string-matching algorithm has been successfully implemented.