Experiment 3.3

Student Name: Utkarsh Joshi **UID:** 21BCS9158
Branch: BE(CSE) **Section/Group:** 802(A)
Semester: 5th **Date of Performance:**06/11/23
Subject Name: DAA **Subject Code:** 21CSH-311

1. **Aim:**
   Develop ac program and analyze complexity to find all occurences of pattern P in a given string S.

2. **Objective:**
   Develop a concise C program to find all occurrences of a pattern P in a given string S using the brute-force pattern matching algorithm.

3. **Algorithm:**

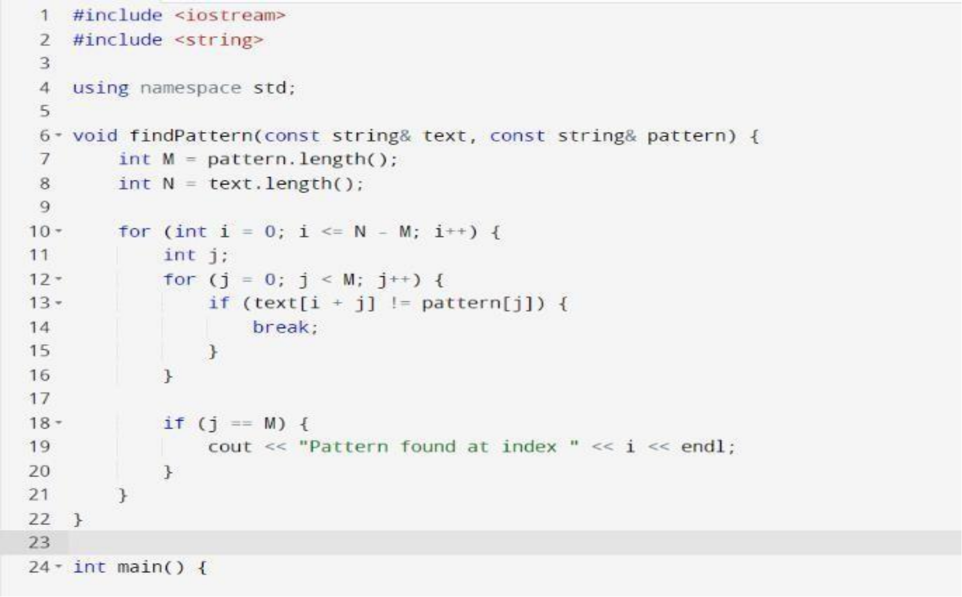
   - Initialize indices i and j to 0.
   - While i is less than or equal to (N - M):
     - While j is less than M and S[i + j] equals P[j, increment j.
     - If j becomes equal to M, a pattern occurrence is found at index i.
     - Increment i and reset j to 0.
   - Repeat until all pattern occurrences in the text are found.

4. **Input/Apparatus Used:**

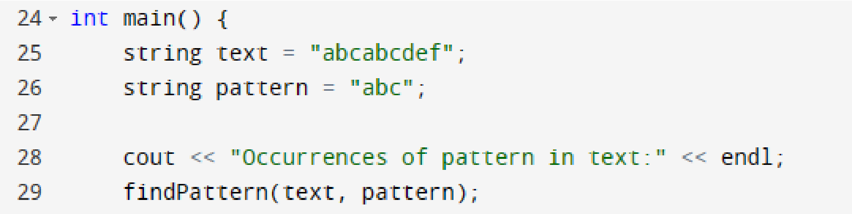
   a. C++ Programming Language

   b. C++ Compiler

6. **Sample Code and Outcome:**

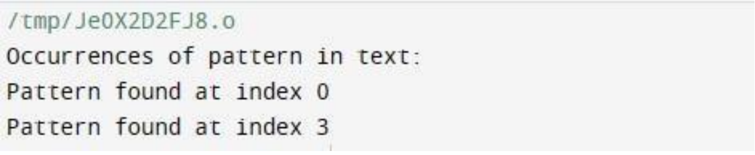


```cpp
#include <iostream>
#include <string>

using namespace std;

void findPattern(const string& text, const string& pattern) {
    int M = pattern.length();
    int N = text.length();

    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) {
            cout << "Pattern found at index " << i << endl;
        }
    }
}

int main() {
```



```cpp
int main() {
    string text = "abcabcdef";
    string pattern = "abc";

    cout << "Occurrences of pattern in text:" << endl;
    findPattern(text, pattern);
```

Outcome:



```
/tmp/Je0X2D2FJ8.o
Occurrences of pattern in text:
Pattern found at index 0
Pattern found at index 3
```

Time complexity:

Time Complexity: The brute-force pattern matching algorithm has a time complexity of O(N*M) in the worst case, where N is the length of the text and M is the length of the pattern.