Q9. b) Implement Naive Searching Algorithm. Analyze its time complexity.

TIME COMPLEXITY: O(m * n)

SPACE COMPLEXITY: O(1)

```
#include <iostream>
#include <queue>
#include <vector>
#include <string>
#include <map>
#include <cmath>

using namespace std;

void search(string & pat, string & txt)
{
    int M = int(pat.size());
    int N = int(txt.size());
    for (int i = 0; i <= N - M; i++)
    {
        int j;
        for (j = 0; j < M; j++)
            if (txt[i + j] != pat[j])
                break;
        if (j == M)
            cout << "Pattern found at index: " << i << endl;
    }
}


int main()
{
    string txt = "ABABDABACDABABCABAB";
    string pat = "ABABCABAB";
    cout << "TEXT: " << txt << endl;
    cout << "PATTERN: " << txt << endl;
    search(pat, txt);
    return 0;
}


//OUTPUT:
//
//TEXT: ABABDABACDABABCABAB
//PATTERN: ABABDABACDABABCABAB
//Pattern found at index: 10
//Program ended with exit code: 0
```