|  |  |
| --- | --- |
| **Brute Force in C++** | |
| #include <iostream>  #include <string>  using namespace std;  void searchPattern(const string& text, const string& pat) {  int m = pat.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] != pat[j]) {  break;  }  }  if (j == m) {  cout << "Pattern found at index " << i << endl;  }  }  }  int main() {  string text = "ababaababbbbabaaa";  string pat = "aa";  cout << "Text: " << text << endl;  cout << "Pattern: " << pat << endl;  searchPattern(text, pat);  return 0;  } | **Input:**   * **Text:** "ababaababbbbabaaa" * **Pattern:** "aa" * **Length of pattern (m):** 2 * **Length of text (n):** 17   **Dry Run Table:**  We'll loop from i = 0 to i = n - m = 15. We check every substring of length 2 and compare with "aa".   | **i** | **text[i..i+1]** | **Matches Pattern?** | | --- | --- | --- | | 0 | ab | ❌ | | 1 | ba | ❌ | | 2 | ab | ❌ | | 3 | ba | ❌ | | 4 | aa | ✅ | | 5 | ab | ❌ | | 6 | bb | ❌ | | 7 | bb | ❌ | | 8 | bb | ❌ | | 9 | bb | ❌ | | 10 | ba | ❌ | | 11 | aa | ✅ | | 12 | aa | ✅ | | 13 | aa | ✅ | | 14 | aa | ✅ | | 15 | — | (out of bounds) |   **✅ Output:**  Pattern found at index 4  Pattern found at index 11  Pattern found at index 12  Pattern found at index 13  Pattern found at index 14 |
| Text: ababaababbbbabaaa  Pattern: aa  Pattern found at index 4  Pattern found at index 14  Pattern found at index 15 | |