Batch: SY-IT(B3) **Experiment Number:5**

Roll Number:16010423076 **Name:Ritesh Jha**

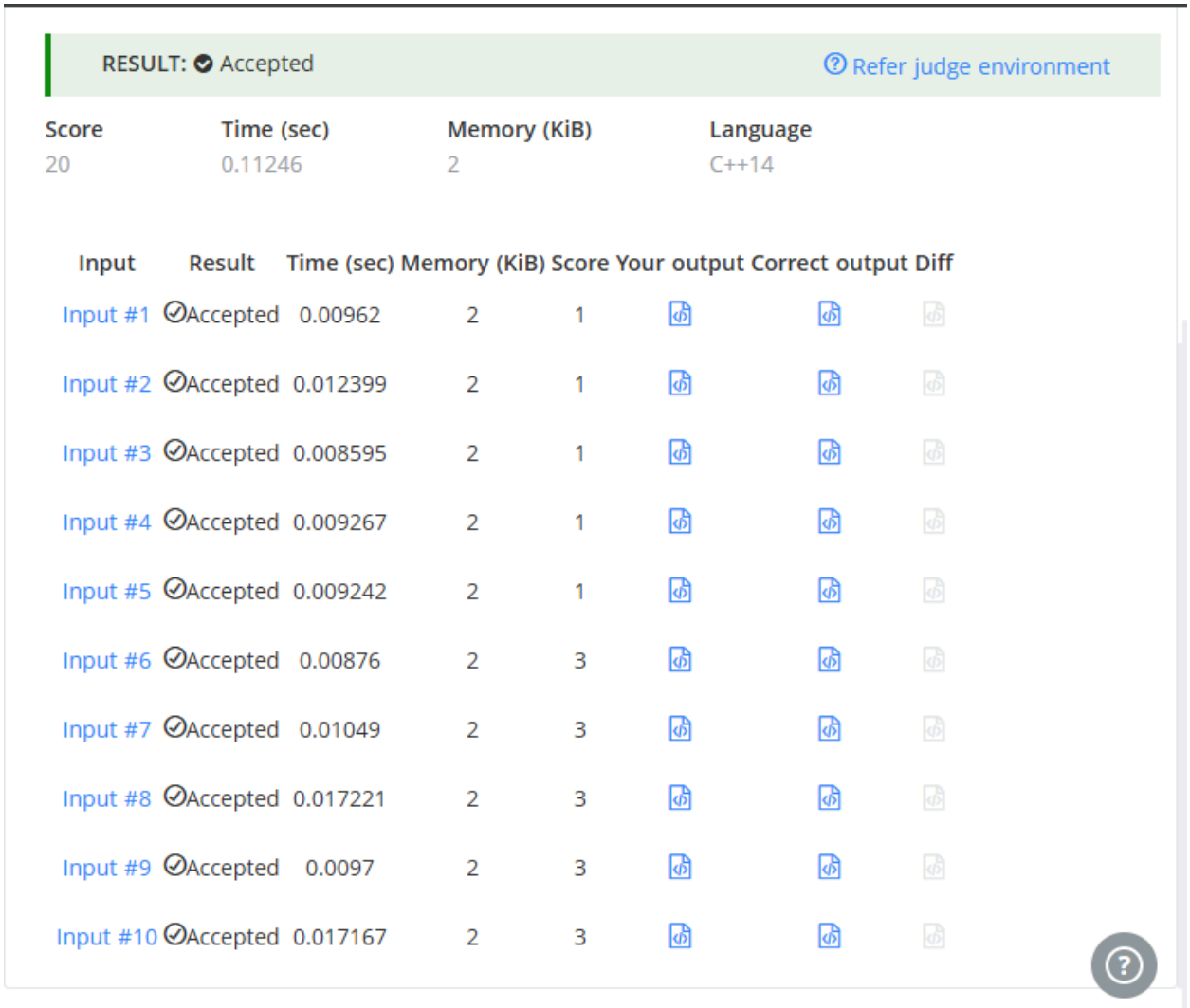
Aim of the Experiment: To study optimization of string handling problem

Program/ Steps:

```
#include<bits/stdc++.h>
using namespace std;

int solve (int N, vector<string> keyboard, string word) {
  // Write your code here
   unordered_map<char, int> row_map;
   for (int i = 0; i < N; i++) {
      for (char c : keyboard[i]) {
         row_map[c] = i;
      }
   }

   int row = row_map[word[0]];
   for (char c : word) {
      if (row_map[c] != row) return 0;
   }

   return 1;
}

int main() {

   ios::sync_with_stdio(0);
   cin.tie(0);
   int N;
```

```
    cin >> N;
    vector<string> keyboard(N);
    for(int i_keyboard = 0; i_keyboard < N; i_keyboard++)
    {
        cin >> keyboard[i_keyboard];
    }
    string word;
    cin >> word;

    int out_;
    out_ = solve(N, keyboard, word);
    cout << out_;
}
```

Output/Result:



RESULT: Accepted — Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20	0.11246	2	C++14

Input	Result	Time (sec)	Memory (KiB)	Score	Your output	Correct output	Diff
Input #1	Accepted	0.00962	2	1			
Input #2	Accepted	0.012399	2	1			
Input #3	Accepted	0.008595	2	1			
Input #4	Accepted	0.009267	2	1			
Input #5	Accepted	0.009242	2	1			
Input #6	Accepted	0.00876	2	3			
Input #7	Accepted	0.01049	2	3			
Input #8	Accepted	0.017221	2	3			
Input #9	Accepted	0.0097	2	3			
Input #10	Accepted	0.017167	2	3			

Outcomes:

CO4. Learn effective computation and programming practices for numeric and string operations and computation geometry

Conclusion (based on the Results and outcomes achieved):

From this experiment, I learned how to optimize string handling by efficiently mapping characters to their respective rows on a keyboard. The program checks if all characters in a given word belong to the same row, improving the processing of string-based input. This helped me understand how to use unordered maps for quick lookups and how to implement efficient logic for checking conditions in strings. The experiment also improved my ability to write clean and optimized C++ code for solving real-world problems related to string operations.

References:

1. https://www.hackerearth.com/practice/algorithms/string-algorithm/string-searching/practice-problems/algorithm/random-keyboard-7fac53bc/
2. T.H. Coreman ,C.E. Leiserson,R.L. Rivest, and C. Stein, " Introduction to algorithms", 3rd Edition 2009, Prentice Hall India Publication
3. Antti Laaksonen, “Guide to Competitive Programming”,Springer,2018
4. Gayle Laakmann McDowell,” Cracking the Coding Interview”,CareerCup LLC,2015
5. Steven S. Skiena Miguel A. Revilla,”Programming challenges, The Programming Contest Training Manual”, Springer, 2006
6. Antti Laaksonen, “Competitive Programmer’s Handbook”, Hand book, 2018
7. Steven Halim and Felix Halim, “Competitive Programming 3: The Lower Bounds of Programming Contests”, Handbook for ACM ICPC