

LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
Semester II tahun 2020/2021

Penyelesaian Cryptarithmic dengan Algoritma Brute Force

Faris Hasim Syauqi
13519050

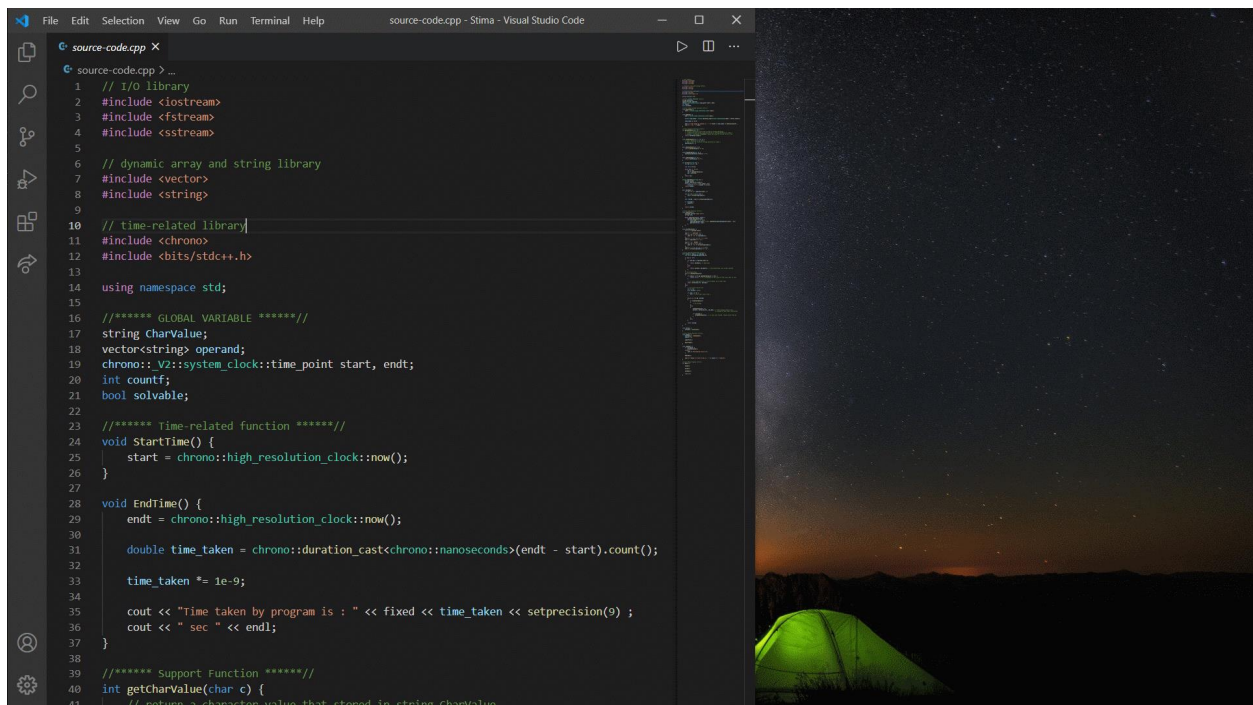
A. Algoritma

Berikut adalah langkah-langkah yang dilakukan dalam penyelesaian persoalan cryptarithmic dengan menggunakan Algoritma Brute Force:

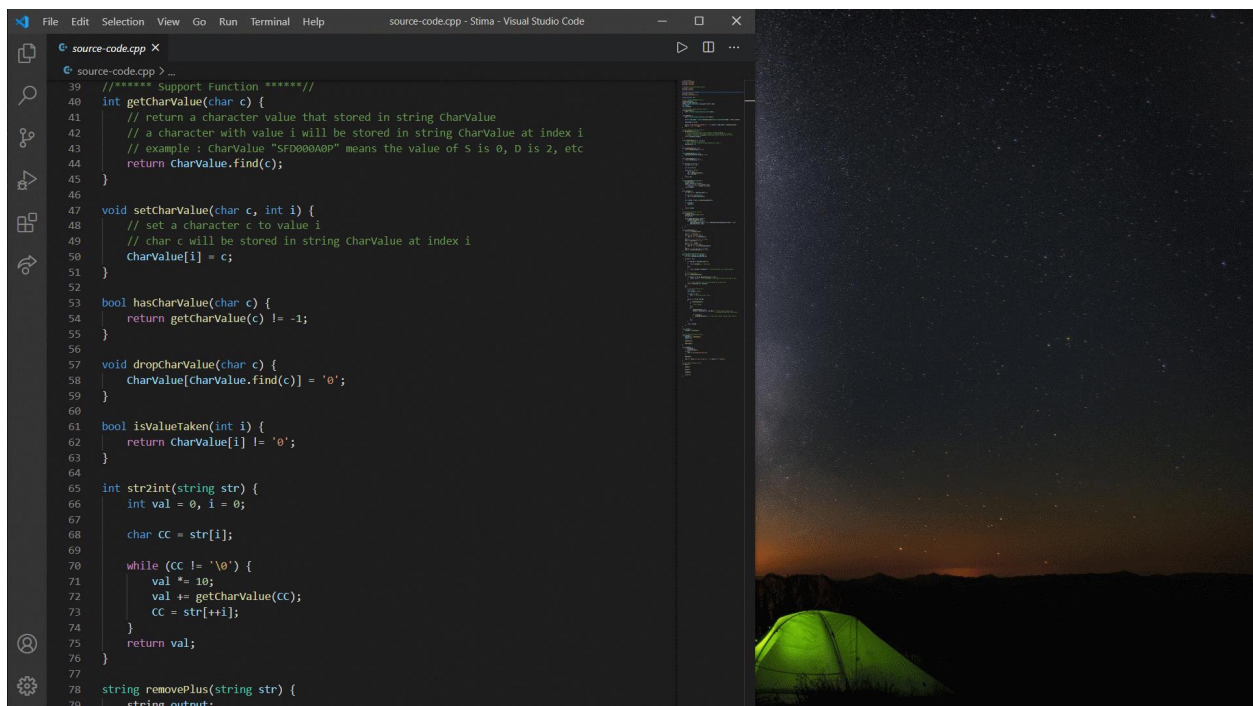
1. Persoalan cryptarithmic akan diselesaikan secara rekursif dimulai dengan beri nilai 1 untuk karakter pertama pada operand pertama. Kemudian geser ke karakter selanjutnya.
2. Basis : terjadi ketika semua huruf pada semua operand (termasuk dalam hal ini kata yang hasil penjumlahan) sudah di-assign dengan suatu nilai, yaitu ketika karakter terakhir pada kata terakhir sudah di-assign dengan suatu nilai, maka akan dilakukan pemeriksaan penjumlahan, setiap huruf akan diganti dengan angka yang di-assign.
Jika Pemeriksaan penjumlahan bernilai benar, maka solusi ditemukan, jika tidak maka mundur satu langkah ke karakter sebelumnya.
3. Rekurens : terjadi selama belum mencapai karakter terakhir pada kata terakhir. Pada kasus ini ada tiga kemungkinan
 - 3.a) Jika semua karakter pada kata/operand saat ini sudah di-assign dengan nilai, maka lanjut ke karakter pertama pada kata/operand selanjutnya.
 - 3.b) Jika karakter saat ini sudah memiliki nilai (karena huruf yang sama sudah muncul sebelumnya), periksa apakah karakter saat ini merupakan awal kata dan apakah nilai karakternya adalah 0. Jika iya maka mundur satu langkah dan huruf sebelumnya naik 1. Selain kasus itu maka maju ke karakter selanjutnya.
 - 3.c) Jika karakter saat ini belum memiliki nilai, maka beri nilai terkecil yang tersedia untuk huruf tersebut, tersedia dalam artian rentang [0,9] yang belum diambil oleh huruf lain. Kemudian lanjut ke karakter selanjutnya. Jika berhasil maka solusi ditemukan, jika tidak maka karakter ini di-assign dengan nilai yang lebih besar terkecil yang tersedia, kemudian lanjut ke karakter selanjutnya. Jika tidak ada satupun nilai yang berujung dengan solusi, maka mundur satu langkah ke karakter sebelumnya.

B. Source Code Program

Berikut adalah source-code program sebagai implementasi dari algoritma di atas



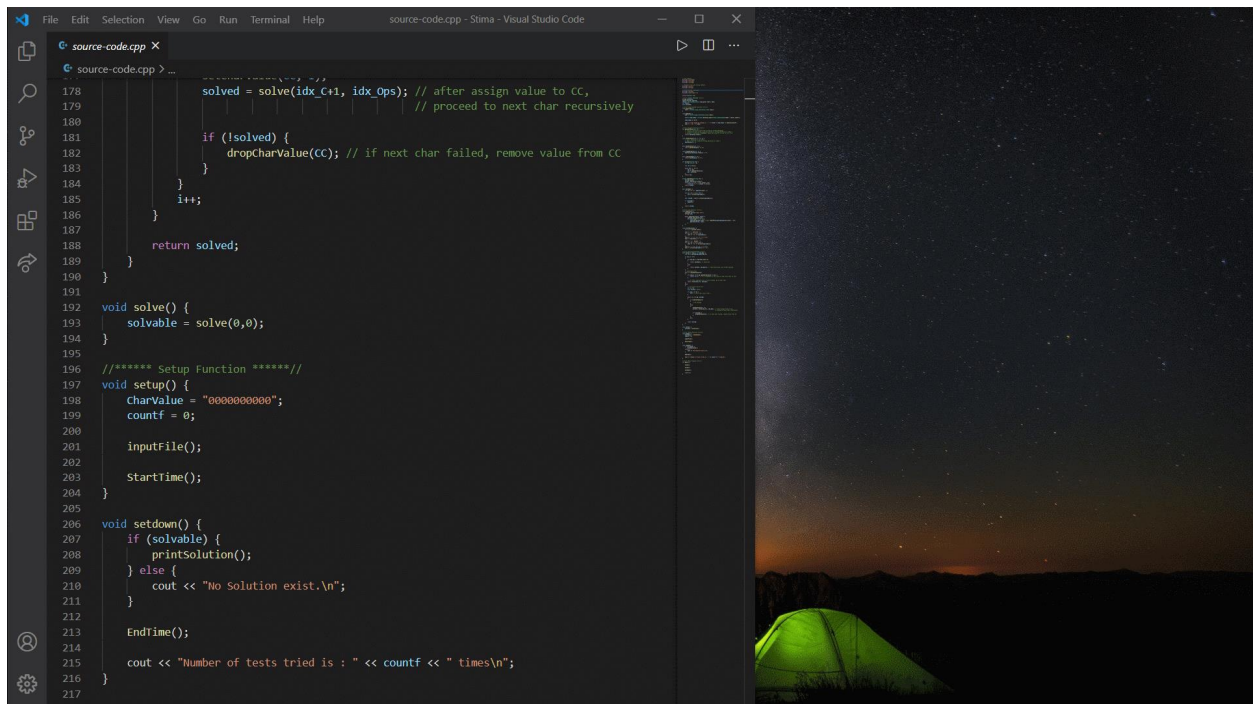
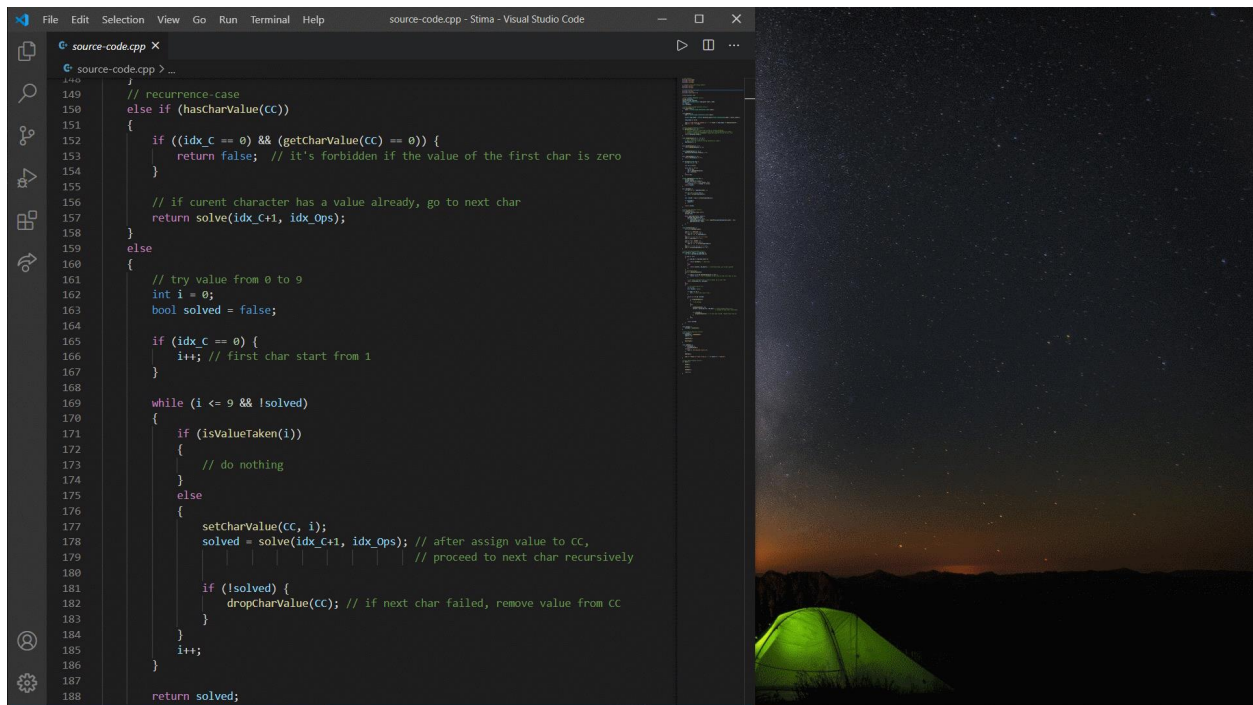
```
1 // I/O library
2 #include <iostream>
3 #include <fstream>
4 #include <sstream>
5
6 // dynamic array and string library
7 #include <vector>
8 #include <string>
9
10 // time-related library
11 #include <chrono>
12 #include <bits/stdc++.h>
13
14 using namespace std;
15
16 //***** GLOBAL VARIABLE *****/
17 string CharValue;
18 vector<string> operand;
19 chrono::V2::system_clock::time_point start, endt;
20 int countf;
21 bool solvable;
22
23 //***** Time-related function *****/
24 void StartTime() {
25     start = chrono::high_resolution_clock::now();
26 }
27
28 void EndTime() {
29     endt = chrono::high_resolution_clock::now();
30
31     double time_taken = chrono::duration_cast<chrono::nanoseconds>(endt - start).count();
32
33     time_taken *= 1e-9;
34
35     cout << "Time taken by program is : " << fixed << time_taken << setprecision(9) ;
36     cout << " sec " << endl;
37 }
38
39 //***** Support Function *****/
40 int getCharValue(char c) {
41     // return a character value that stored in string CharValue
```

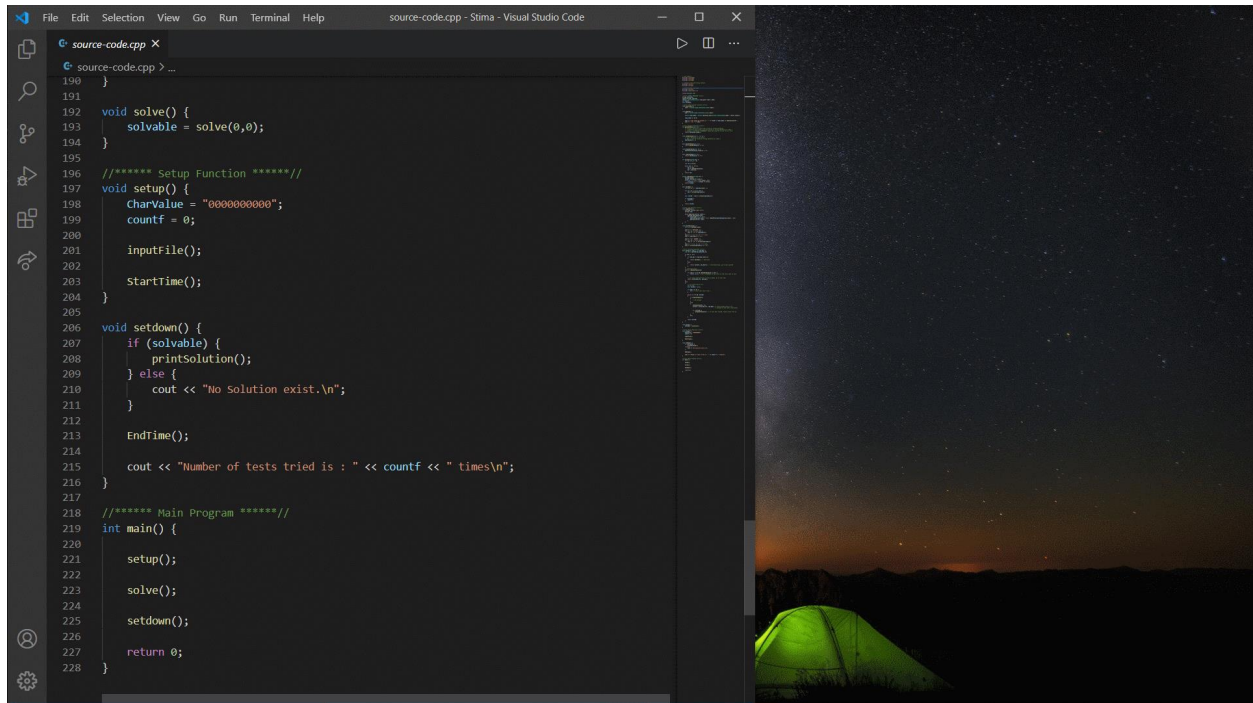


```
39 //***** Support Function *****/
40 int getCharValue(char c) {
41     // return a character value that stored in string CharValue
42     // a character with value i will be stored in string CharValue at index i
43     // example : CharValue "SFDDDDABP" means the value of S is 0, D is 2, etc
44     return CharValue.find(c);
45 }
46
47 void setCharValue(char c, int i) {
48     // set a character c to value i
49     // char c will be stored in string CharValue at index i
50     CharValue[i] = c;
51 }
52
53 bool hasCharValue(char c) {
54     return getCharValue(c) != -1;
55 }
56
57 void dropCharValue(char c) {
58     CharValue[CharValue.find(c)] = '0';
59 }
60
61 bool isValueTaken(int i) {
62     return CharValue[i] != '0';
63 }
64
65 int str2int(string str) {
66     int val = 0, i = 0;
67
68     char CC = str[i];
69
70     while (CC != '\0') {
71         val *= 10;
72         val += getCharValue(CC);
73         CC = str[++i];
74     }
75     return val;
76 }
77
78 string removePlus(string str) {
79     string output;
```

```
File Edit Selection View Go Run Terminal Help source-code.cpp - Stima - Visual Studio Code
source-code.cpp X
source-code.cpp > ...
77
78 string removePlus(string str) {
79     string output;
80     output.reserve(str.size());
81     for(size_t i = 0; i < str.size(); ++i)
82         if(str[i] != '+') output += str[i];
83     return output;
84 }
85
86 bool testSum() {
87     int sum = 0, n = operand.size(), i;
88
89     for (i = 0; i < n-1; i++) {
90         sum += str2int(operand[i]);
91     }
92
93     bool solved = (sum == str2int(operand[i]));
94
95     if (!solved) {
96         countf++;
97     }
98
99     return solved;
100 }
101
102 //***** I/O Function *****/
103 void inputFile() {
104     ifstream infile("input.txt");
105     string line;
106
107     while (getline(infile, line)) {
108         operand.push_back(line);
109         if (line.find('+') != -1) {
110             operand[operand.size() - 1] = removePlus(operand[operand.size() - 1]);
111             getline(infile, line);
112         }
113     }
114 }
115
116 void printSolution() {
117     int i,n = operand.size();
```

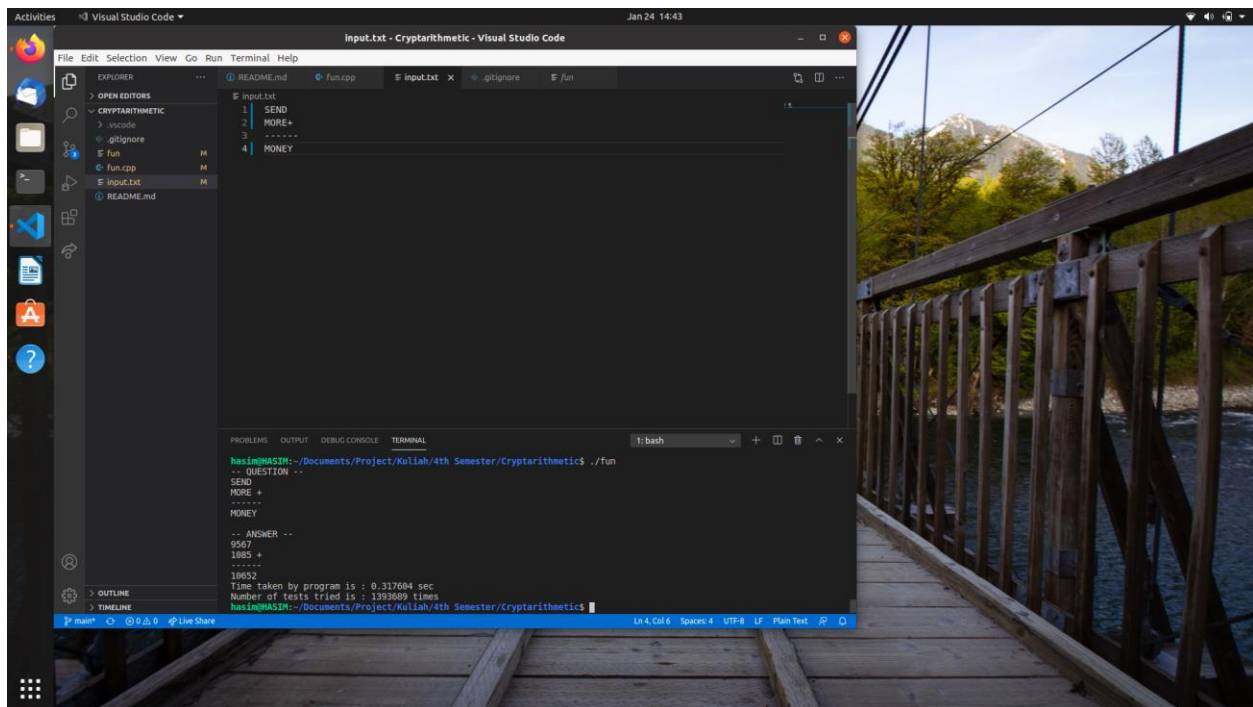
```
File Edit Selection View Go Run Terminal Help source-code.cpp - Stima - Visual Studio Code
source-code.cpp X
source-code.cpp > ...
115
116 void printSolution() {
117     int i,n = operand.size();
118
119     cout << "--- QUESTION ---";
120     for (i = 0; i < n-1; i++) {
121         cout << '\n' << operand[i];
122     }
123     cout << " + " << '\n' << "-----\n";
124     cout << operand[i] << '\n';
125
126     cout << "\n-- ANSWER --";
127     for (i = 0; i < n-1; i++) {
128         cout << '\n' << str2int(operand[i]);
129     }
130     cout << " + " << '\n' << "-----\n";
131     cout << str2int(operand[i]) << '\n';
132 }
133
134 //***** Solving Algorithm *****/
135 bool solve(int idx_C, int idx_Ops) {
136     char CC = operand[idx_Ops][idx_C];
137
138     if (CC == '\0')
139     {
140         if (idx_Ops == operand.size()-1)
141         {
142             return testSum(); // base-case
143         }
144         else
145         {
146             return solve(0, idx_Ops+1); // recurrence-case, go to next operand
147         }
148     }
149     // recurrence-case
150     else if (hasCharValue(CC))
151     {
152         if ((idx_C == 0) && (getCharValue(CC) == 0)) {
153             return false; // it's forbidden if the value of the first char is zero
154         }
155     }
```

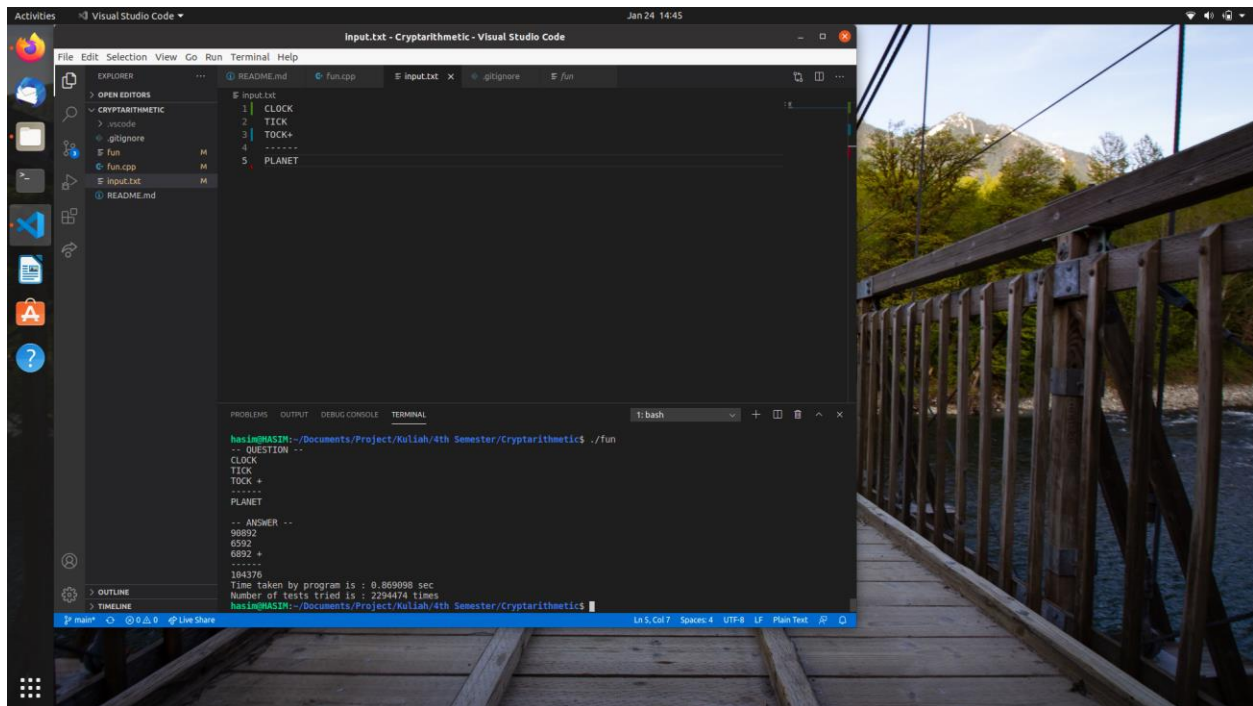
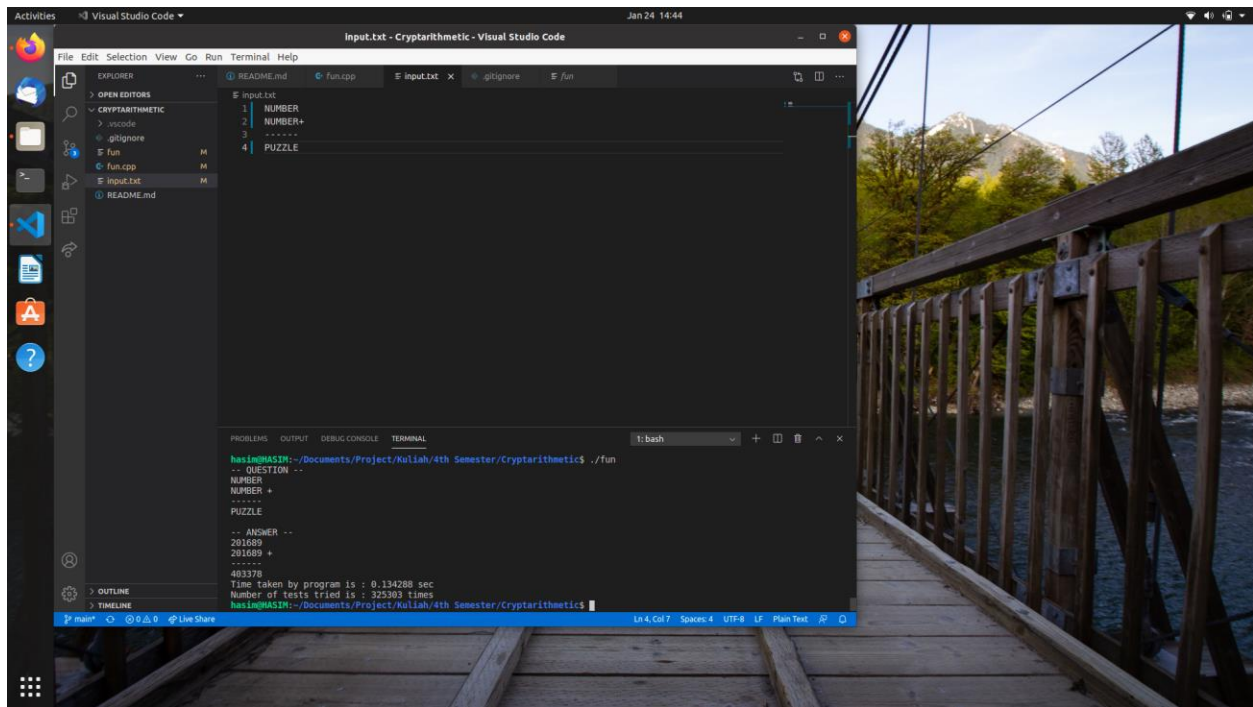



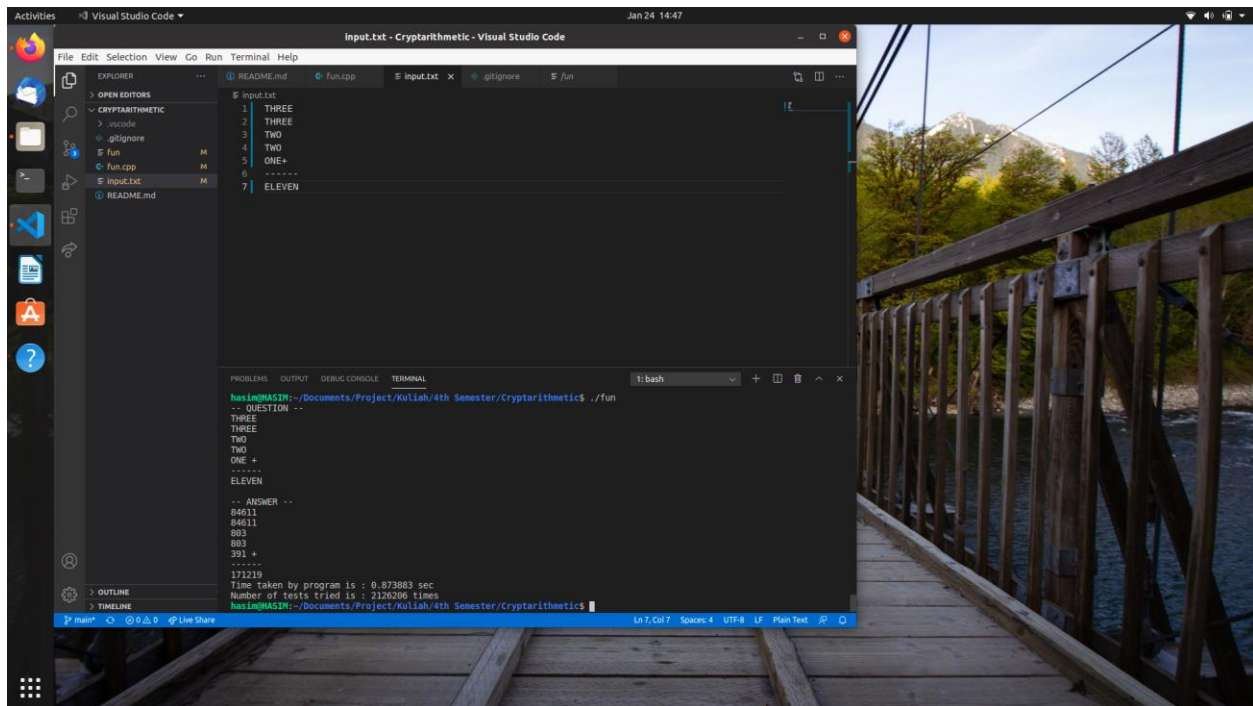
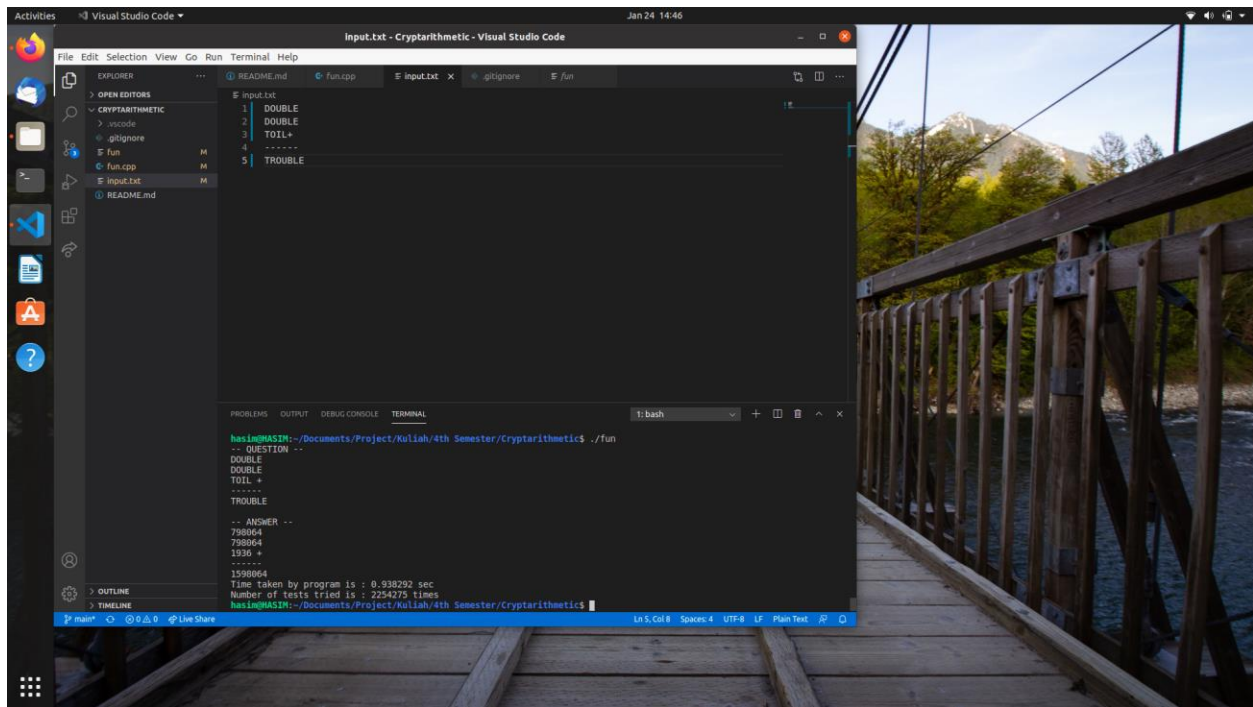


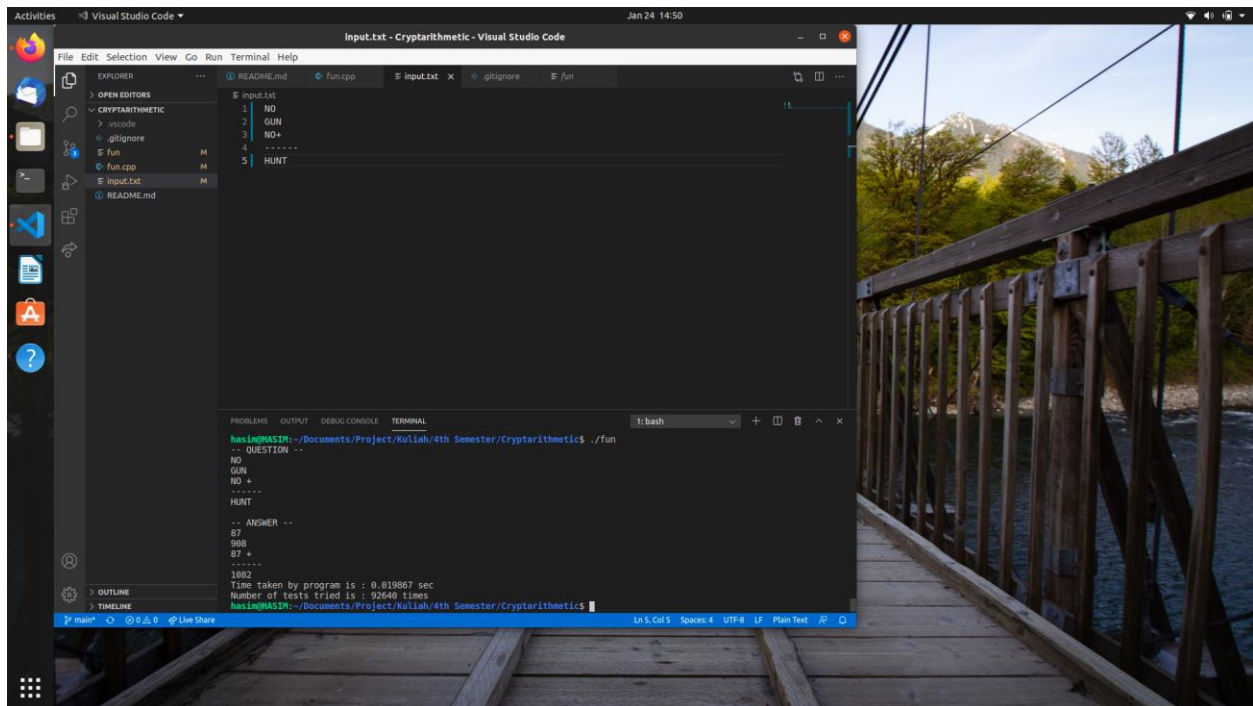
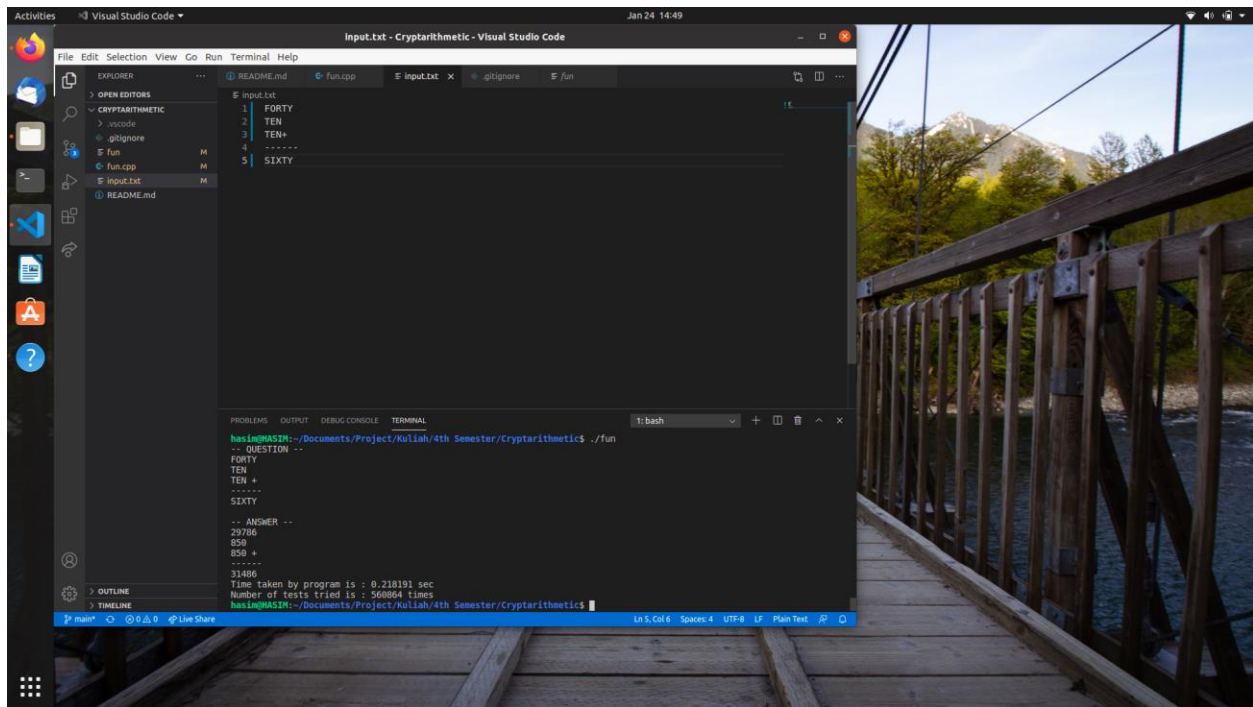
C. Screenshot Percobaan

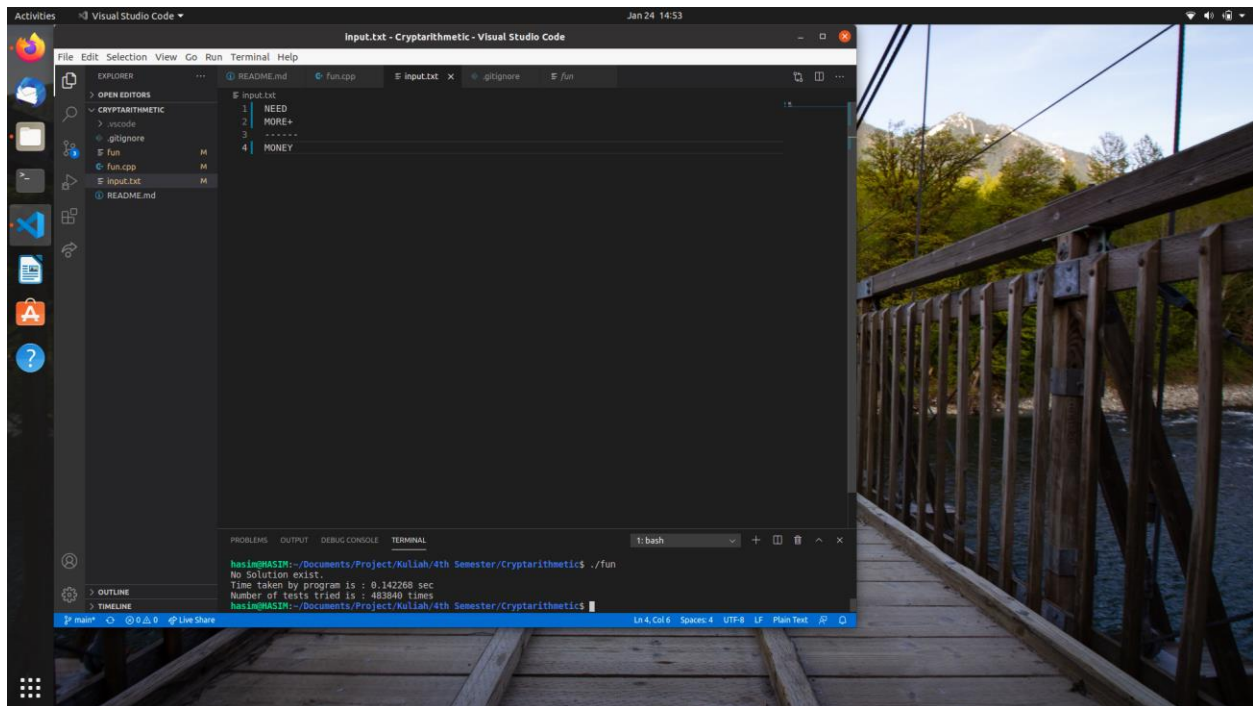
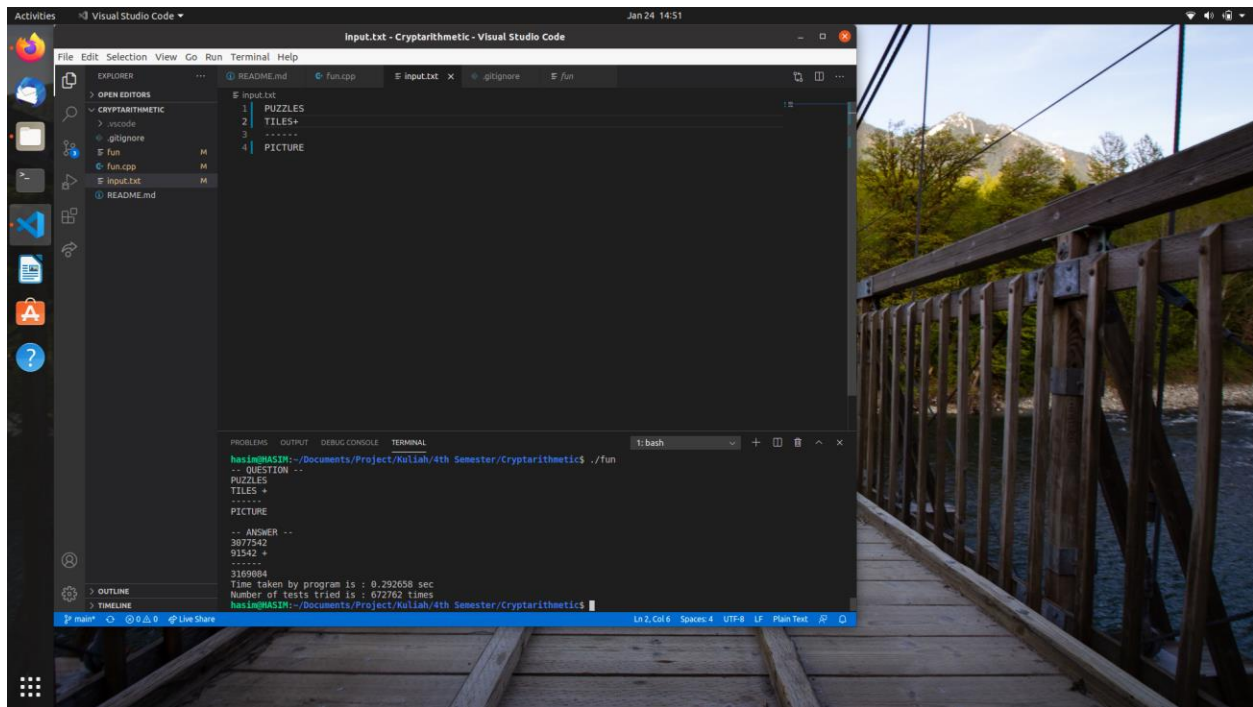
Berikut adalah beberapa foto hasil percobaan dari program.

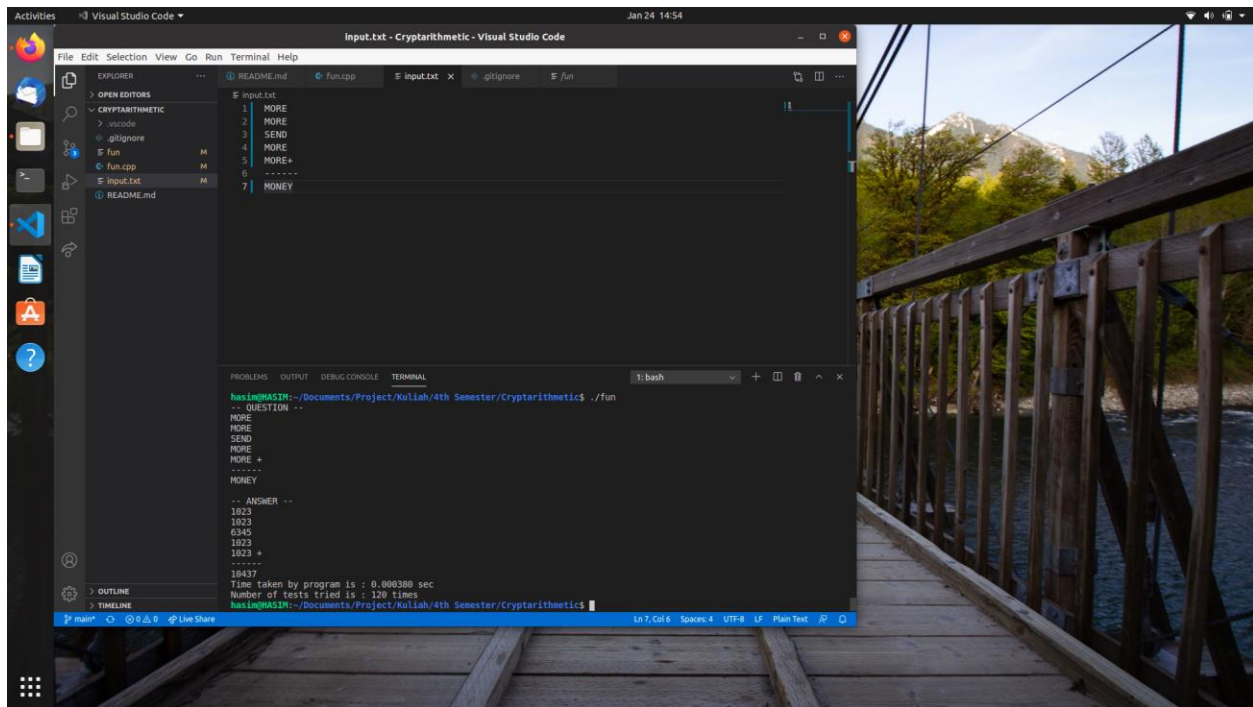












D. Alamat Kode Program

Program dapat digunakan dari alamat berikut : <https://github.com/farishasim/Cryptarithmetic>