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

# Penyelesaian Cryptarithmetic dengan Algoritma Brute Force

Faris Hasim Syauqi
13519050

## A. Algoritma

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

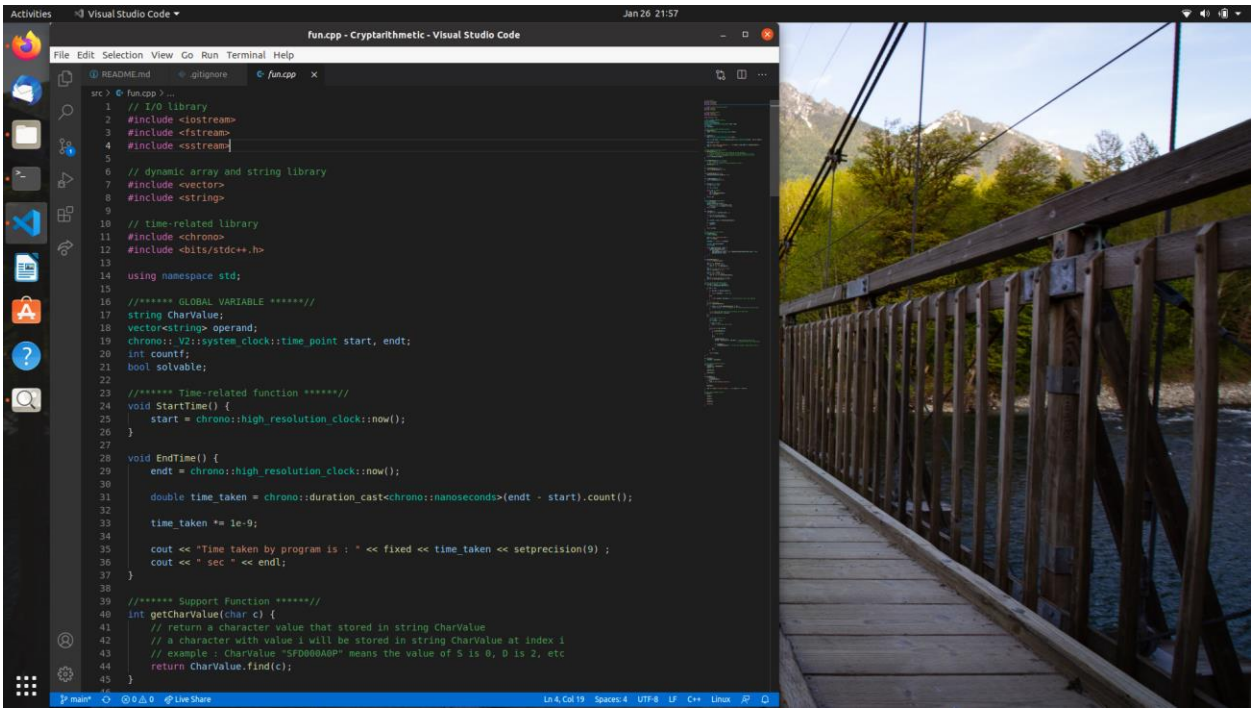
1. Persoalan cyrptarithmetic akan diselesaikan secara rekursif dimulai dengan beri nilai 1 untuk karakter pertama pada operand pertama. Kemudian geser ke karakter selanjutnya.

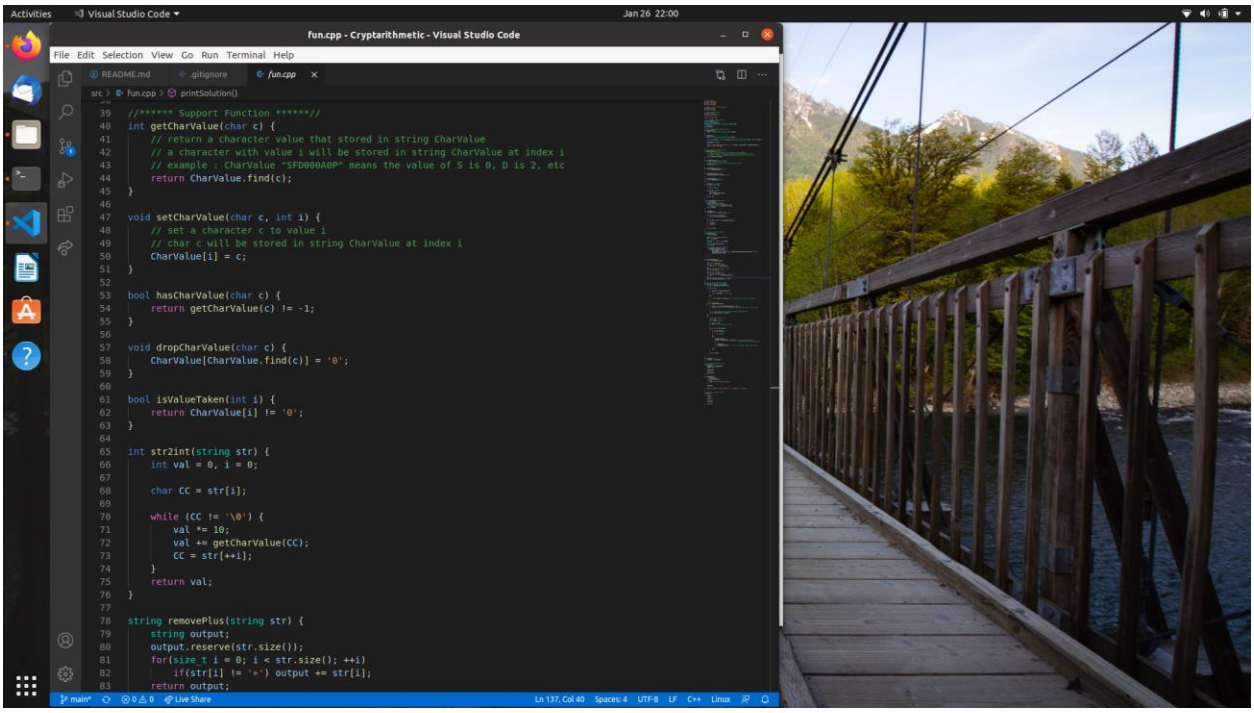
2. Basis : terjadi ketika semua huruf pada semua operand (termasuk dalam hal ini kata yang hasil penjumlahan) sudah di-assign dengan suatu nilai, yaitu ketika karakter terakhir pada kata terakhir sudah di-assign dengan suatu nilai, maka akan dilakukan pemeriksaan penjumlahan, setiap huruf akan diganti dengan angka yang di-assign.
   Jika Pemeriksaan penjumlahan bernilai benar, maka solusi ditemukan, jika tidak maka mundur satu langkah ke karakter sebelumnya.

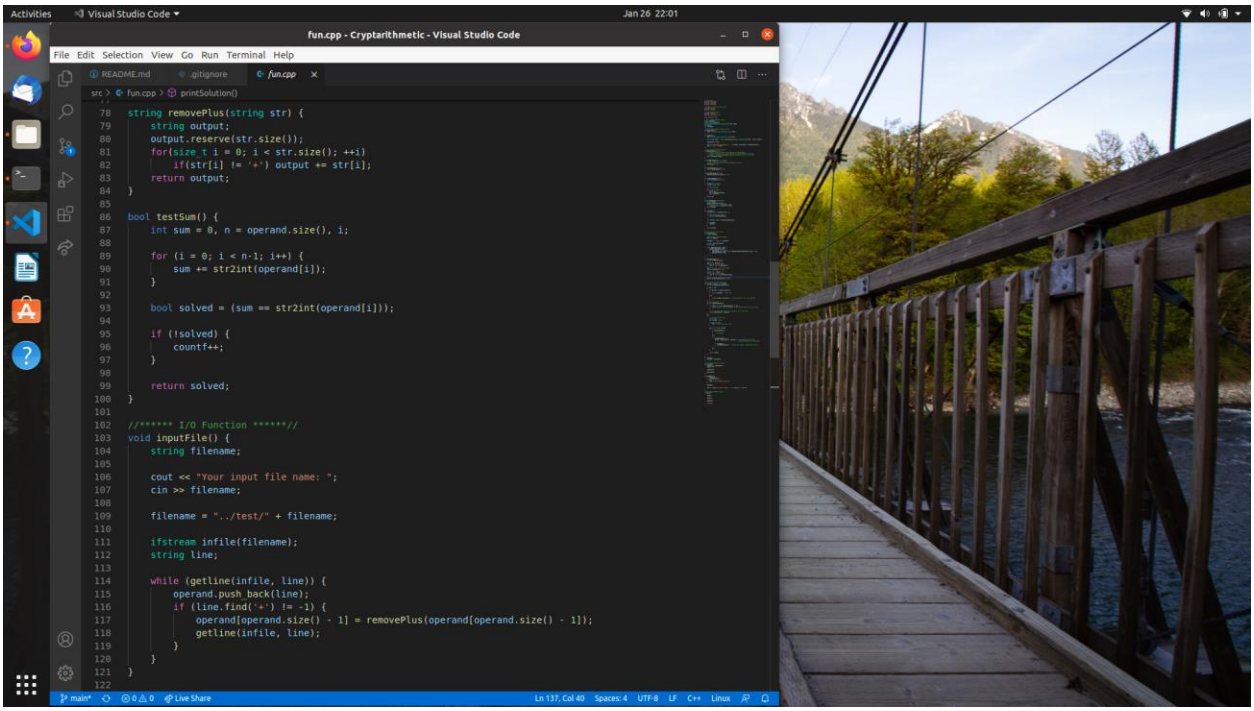
3. Rekurens : terjadi selama belum mencapai karakter terkahir pada kata terakhir. Pada kasus ini ada tiga kemungkinan
   3.a) Jika semua karakter pada kata/operand saat ini sudah di-assign dengan nilai, maka lanjut ke karakter pertama pada kata/operand selanjutnya.
   3.b) Jika karakter saat ini sudah memliki nilai (karena huruf yang sama sudah muncul sebelumnya), periksa apakah karakter saat ini merupakan awal kata dan apakah nilai karakternya adalah 0. Jika iya, karena karakter pertama tidak boleh bernilai nol maka mundur satu langkah ke karakter sebelumnya. 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.

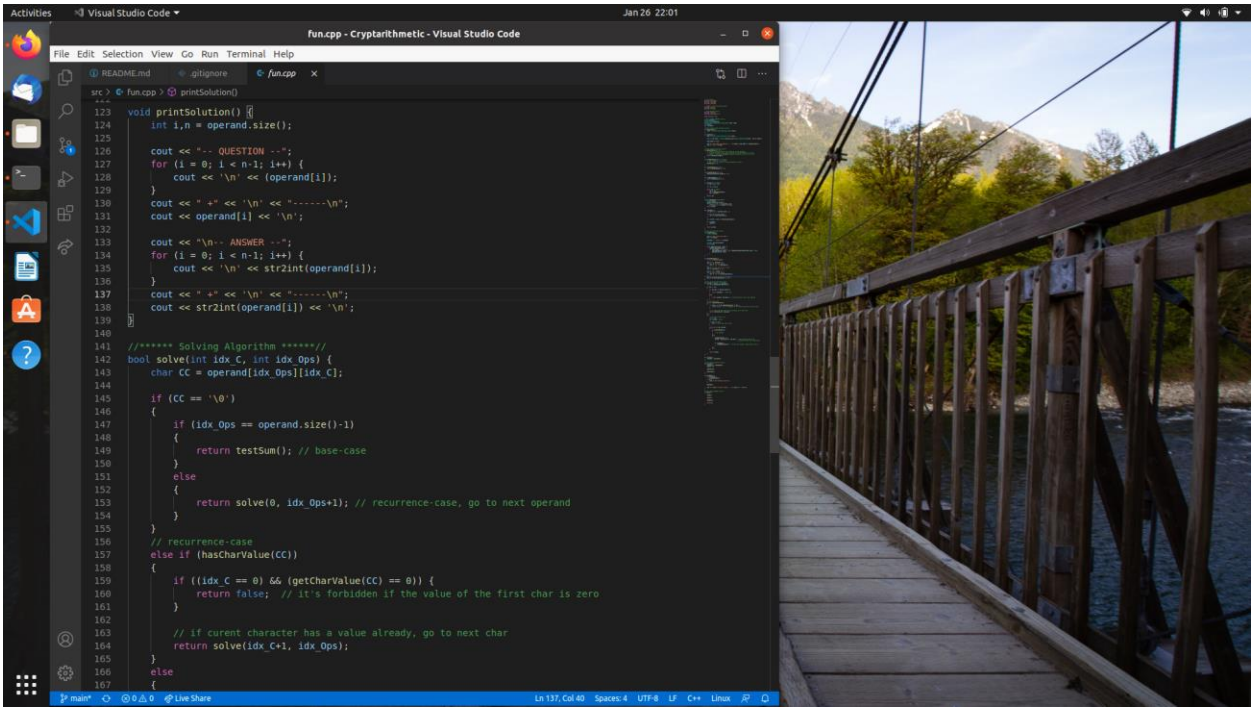
## B. *Source Code Program*

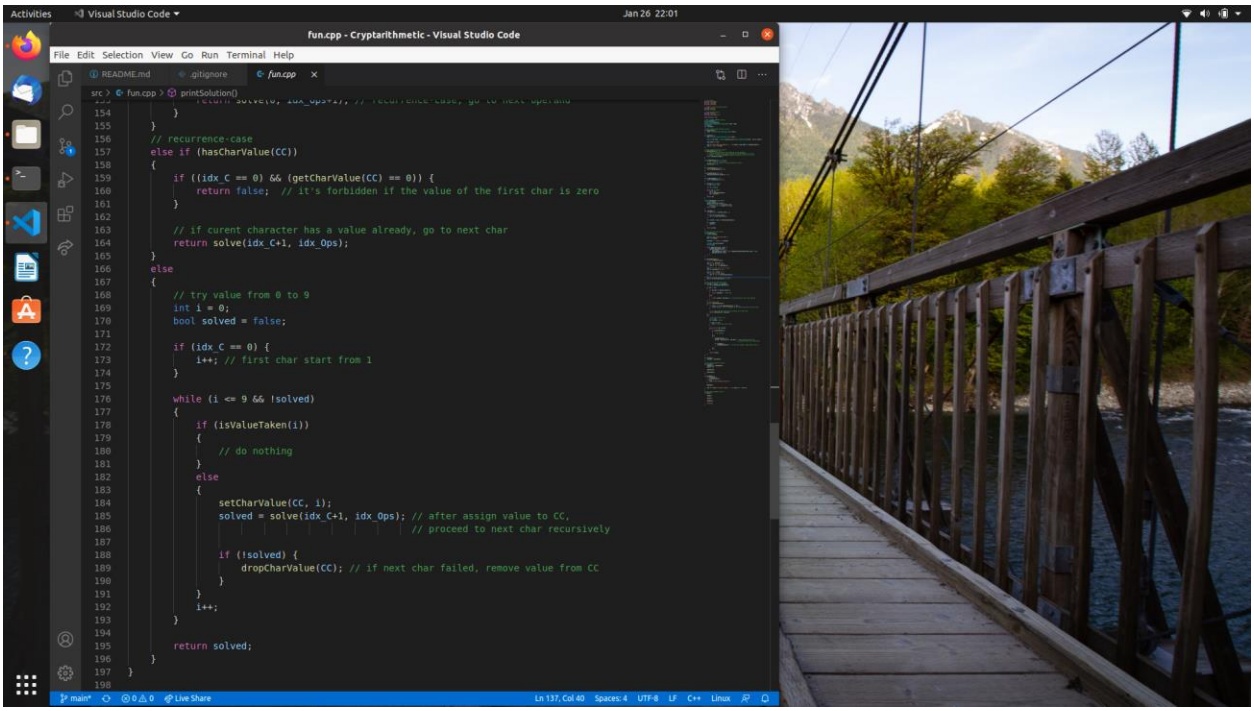
Berikut adalah source-code program sebagai implementasi dari algoritma di atas.

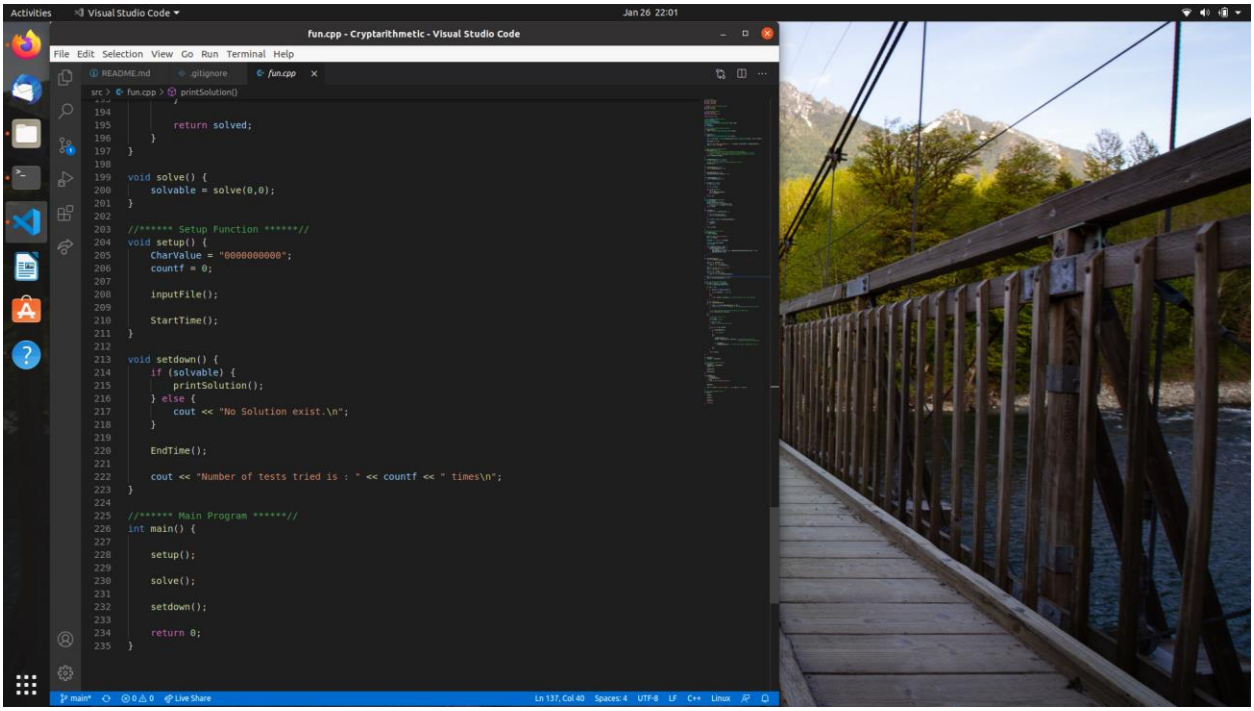



Activities
Visual Studio Code
Jan 26 22:01
fun.cpp - Cryptarithmetic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
README.md
.gitignore
fun.cpp
src > fun.cpp > printSolution()
string removePlus(string str) {
string output;
output.reserve(str.size());
for(size_t i = 0; i < str.size(); ++i)
if(str[i] != '+') output += str[i];
return output;
}
bool testSum() {
int sum = 0, n = operand.size(), i;
for (i = 0; i < n-1; i++) {
sum += str2int(operand[i]);
}
bool solved = (sum == str2int(operand[i]));
if (!solved) {
countf++;
}
return solved;
}
//****** I/O Function ******//
void inputFile() {
string filename;
cout << "Your input file name: ";
cin >> filename;
filename = "../test/" + filename;
ifstream infile(filename);
string line;
while (getline(infile, line)) {
operand.push_back(line);
if (line.find('+') != -1) {
operand[operand.size() - 1] = removePlus(operand[operand.size() - 1]);
getline(infile, line);
}
}
}
main*
0 0
Live Share
Ln 137, Col 40
Spaces: 4
UTF-8
LF
C++
Linux


Activities
Visual Studio Code
Jan 26 22:01
fun.cpp - Cryptarithmetic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
README.md
.gitignore
fun.cpp
src > fun.cpp > printSolution()
void printSolution() {
int i,n = operand.size();
cout << "-- QUESTION --";
for (i = 0; i < n-1; i++) {
cout << '\n' << (operand[i]);
}
cout << " +" << '\n' << "------\n";
cout << operand[i] << '\n';
cout << "\n-- ANSWER --";
for (i = 0; i < n-1; i++) {
cout << '\n' << str2int(operand[i]);
}
cout << " +" << '\n' << "------\n";
cout << str2int(operand[i]) << '\n';
}
//****** Solving Algorithm ******//
bool solve(int idx_C, int idx_Ops) {
char CC = operand[idx_Ops][idx_C];
if (CC == '\0')
{
if (idx_Ops == operand.size()-1)
{
return testSum(); // base-case
}
else
{
return solve(0, idx_Ops+1); // recurrence-case, go to next operand
}
}
// recurrence-case
else if (hasCharValue(CC))
{
if ((idx_C == 0) && (getCharValue(CC) == 0)) {
return false; // it's forbidden if the value of the first char is zero
}
// if curent character has a value already, go to next char
return solve(idx_C+1, idx_Ops);
}
else
{
main*
0 0
Live Share
Ln 137, Col 40
Spaces: 4
UTF-8
LF
C++
Linux

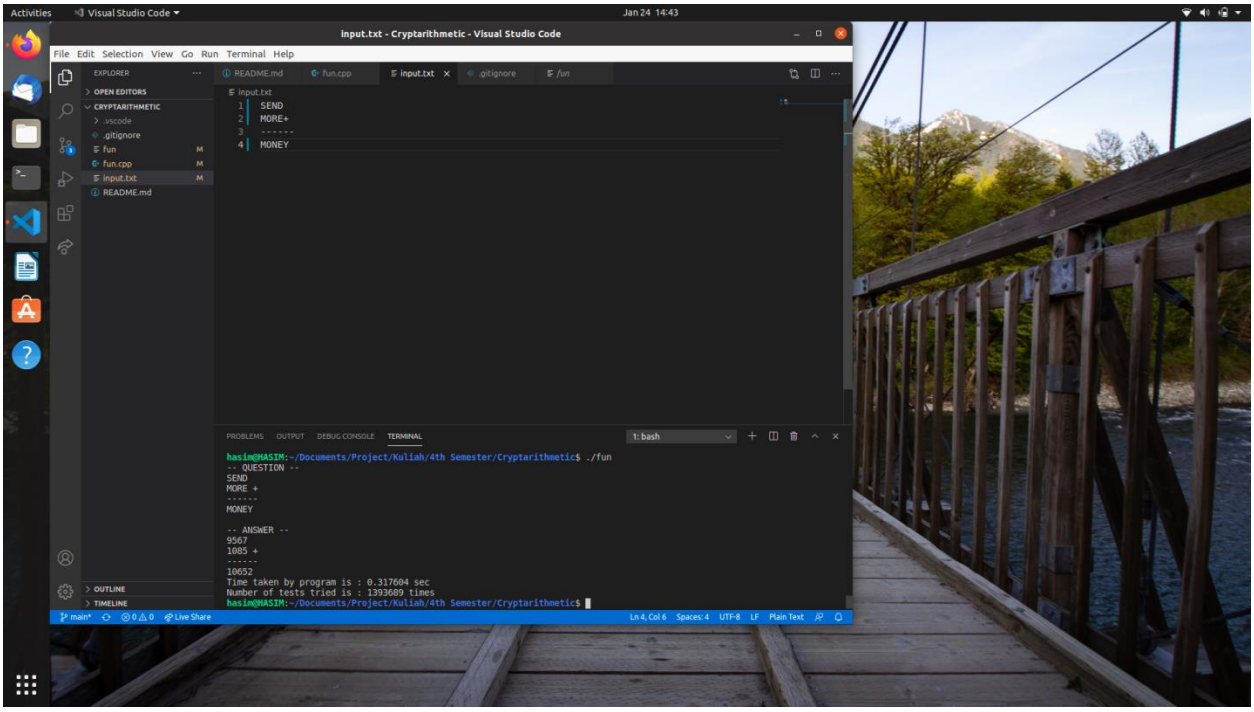

Activities
Visual Studio Code
Jan 26 22:01
fun.cpp - Cryptarithmetic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
README.md
.gitignore
fun.cpp
src > fun.cpp > printSolution()
154 }
155 }
156 // recurrence-case
157 else if (hasCharValue(CC))
158 {
159 if ((idx_C == 0) && (getCharValue(CC) == 0)) {
160 return false; // it's forbidden if the value of the first char is zero
161 }
162
163 // if curent character has a value already, go to next char
164 return solve(idx_C+1, idx_Ops);
165 }
166 else
167 {
168 // try value from 0 to 9
169 int i = 0;
170 bool solved = false;
171
172 if (idx_C == 0) {
173 i++; // first char start from 1
174 }
175
176 while (i <= 9 && !solved)
177 {
178 if (isValueTaken(i))
179 {
180 // do nothing
181 }
182 else
183 {
184 setCharValue(CC, i);
185 solved = solve(idx_C+1, idx_Ops); // after assign value to CC,
186 // proceed to next char recursively
187
188 if (!solved) {
189 dropCharValue(CC); // if next char failed, remove value from CC
190 }
191 }
192 i++;
193 }
194
195 return solved;
196 }
197 }
198
main*
0 0
Live Share
Ln 137, Col 40
Spaces: 4
UTF-8
LF
C++
Linux


Activities
Visual Studio Code
Jan 26 22:01
fun.cpp - Cryptarithmetic - Visual Studio Code
File Edit Selection View Go Run Terminal Help
README.md
.gitignore
fun.cpp
src > fun.cpp > printSolution()
194
195 return solved;
196 }
197 }
198
199 void solve() {
200 solvable = solve(0,0);
201 }
202
203 //****** Setup Function ******//
204 void setup() {
205 CharValue = "0000000000";
206 countf = 0;
207
208 inputFile();
209
210 StartTime();
211 }
212
213 void setdown() {
214 if (solvable) {
215 printSolution();
216 } else {
217 cout << "No Solution exist.\n";
218 }
219
220 EndTime();
221
222 cout << "Number of tests tried is : " << countf << " times\n";
223 }
224
225 //****** Main Program ******//
226 int main() {
227
228 setup();
229
230 solve();
231
232 setdown();
233
234 return 0;
235 }
main*
0 0
Live Share
Ln 137, Col 40
Spaces: 4
UTF-8
LF
C++
Linux

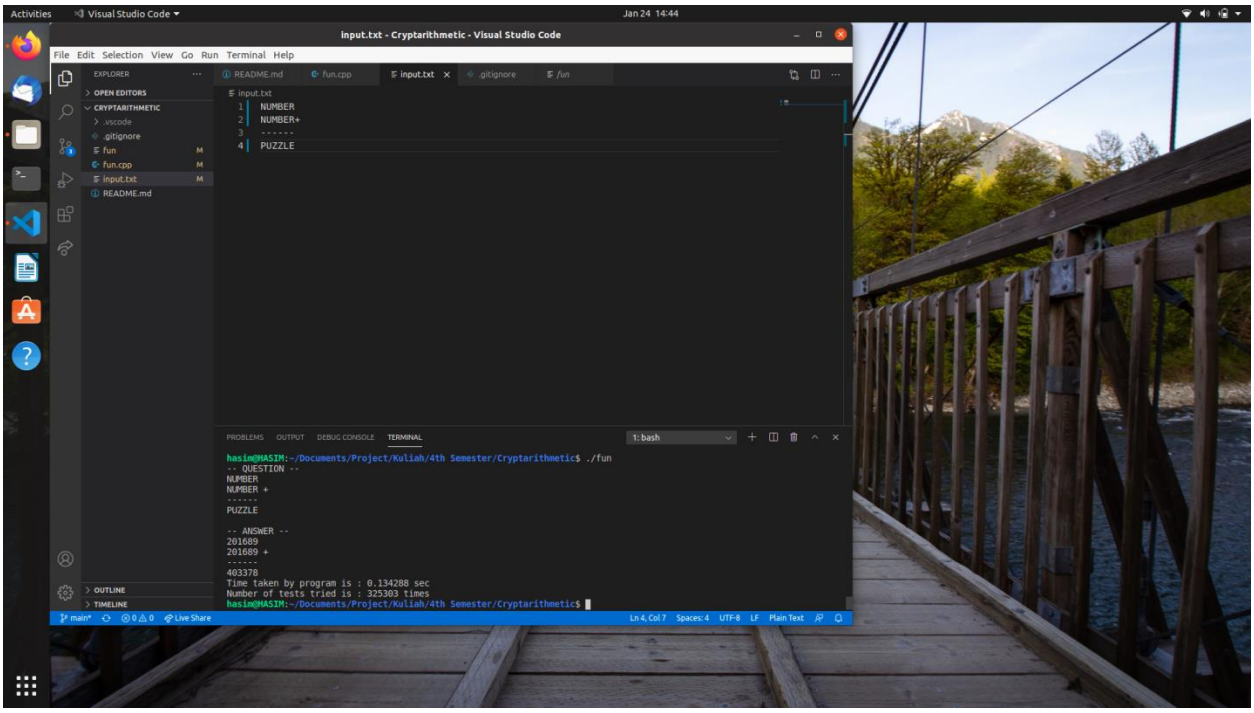
## C. *Screenshot* Percobaan

Berikut adalah beberapa foto hasil percobaan dari program.

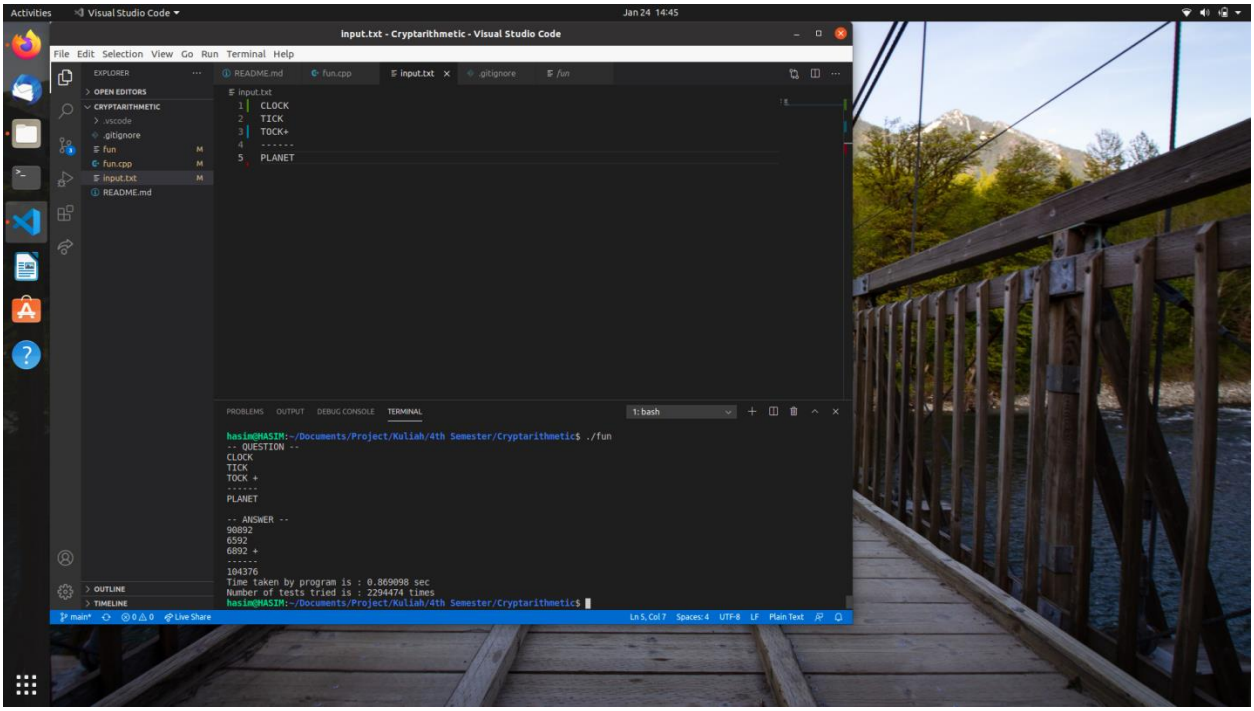
1. SEND + MORE = MONEY



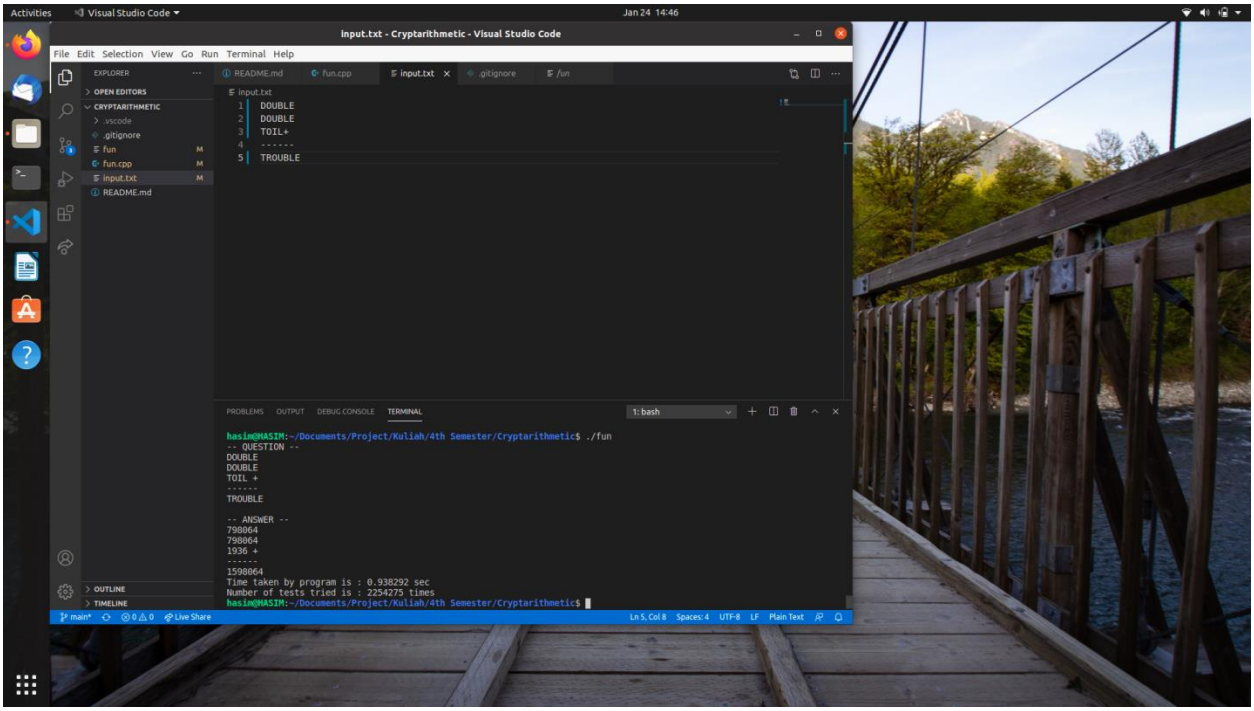

2. NUMBER + NUMBER = PUZZLE



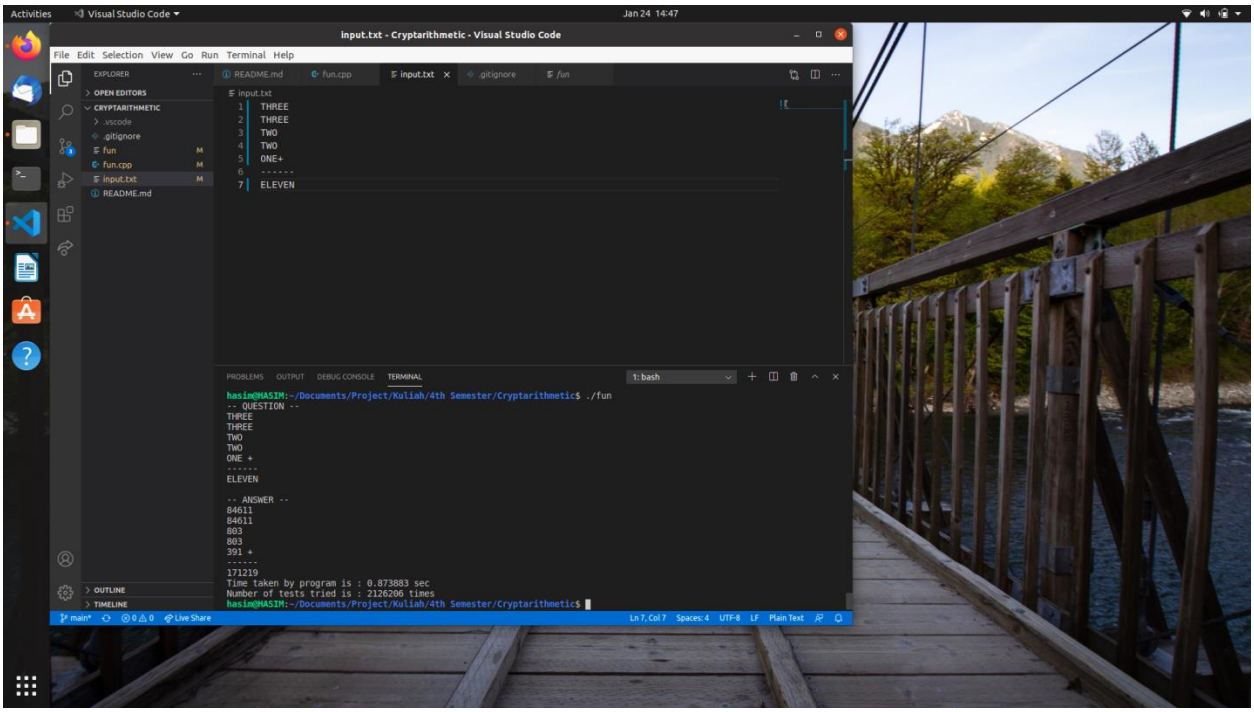
3. CLOCK + TICK + TOCK = PLANET



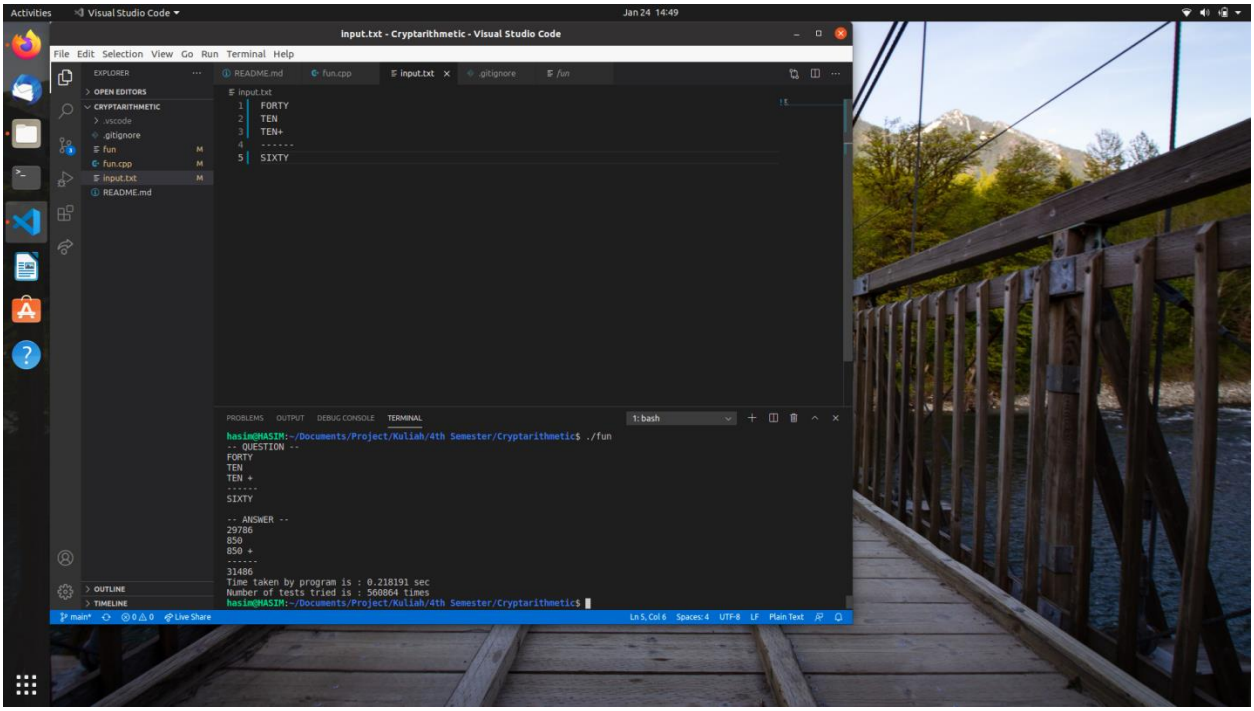

4. DOUBLE + DOUBLE + TOIL = TROUBLE



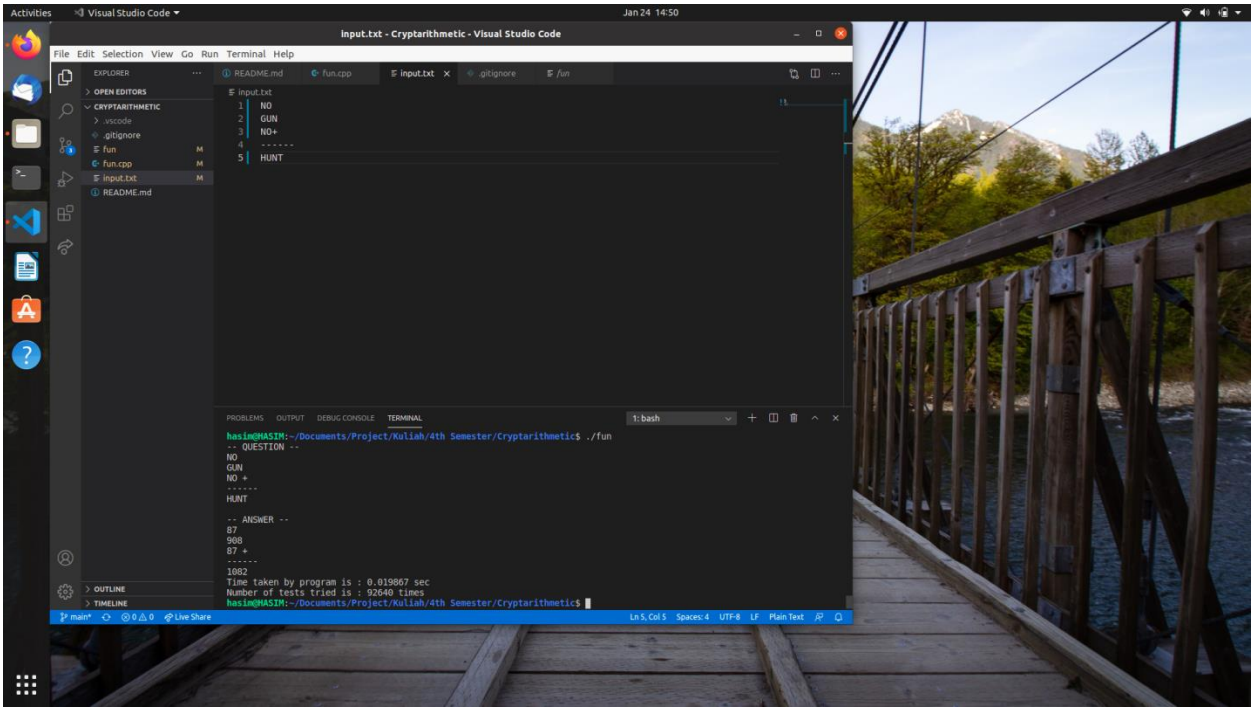
5. THREE + THREE + TWO + TWO + ONE = ELEVEN



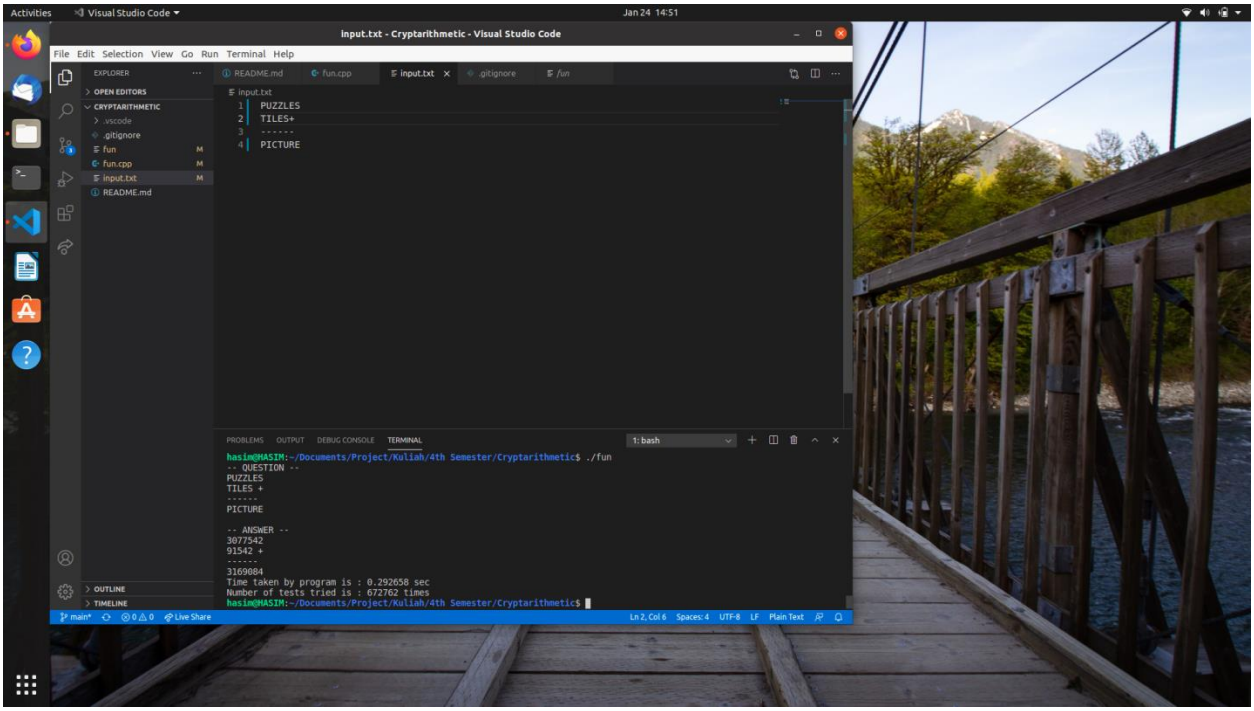

6. FORTY + TEN + TEN = SIXTY



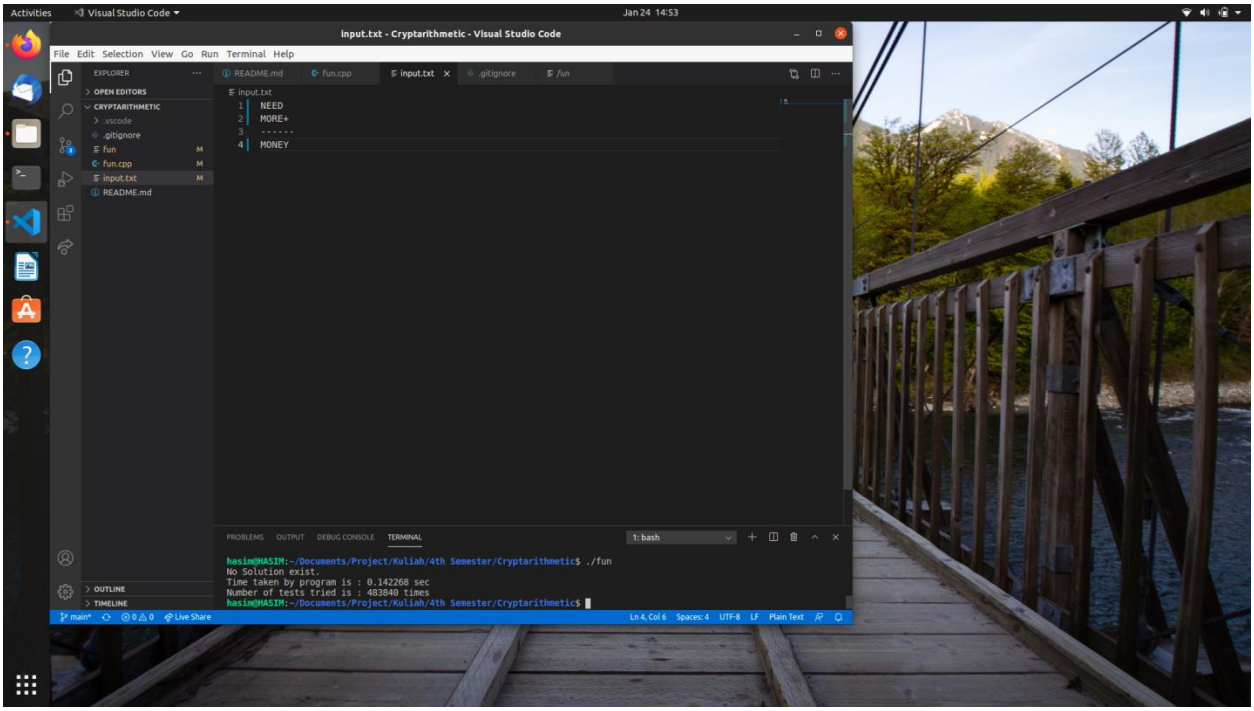
7. NO + GUN + NO = HUNT



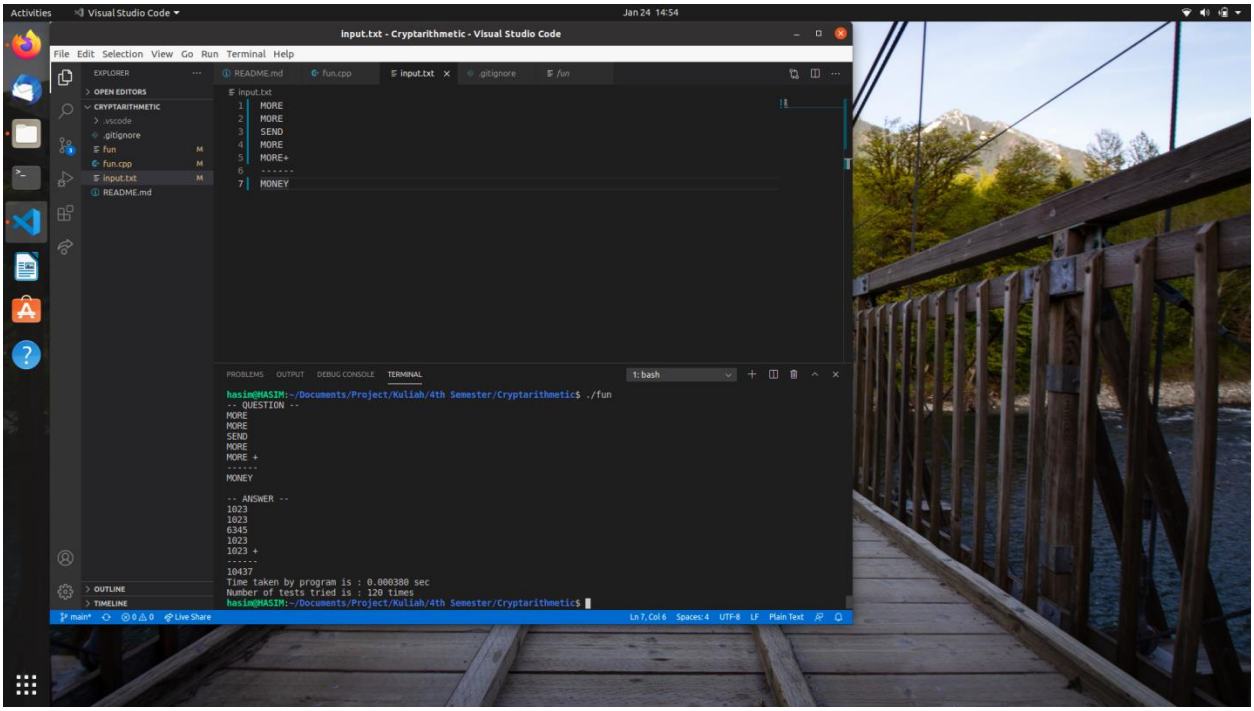

8. PUZZLES + TILES = PICTURE



9. NEED + MORE = MONEY




10. MORE + MORE + SEND + MORE + MORE = MONEY



## D. Alamat Kode Program

Program dapat diunduh dari alamat berikut :

Drive : https://drive.google.com/drive/u/0/folders/1y7u0LIIha5tV8R6sr9vOnMFQqVzRVUf2

(Pilih Tucil1_13519050.zip) atau dapat diunduh dari github pada tautan berikut :

GitHub : https://github.com/farishasim/Cryptarithmetic

## E. Tabel Penilaian

| **Poin** | Ya | Tidak |
|---|---|---|
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | √ | |
| 2. Program berhasil running | √ | |
| 3. Program dapat membaca file masukan dan menuliskan luaran | √ | |
| 4. Solusi cryptarithmetic hanya benar untuk persoalan cryptarihtmetic dengan dua buah operand. | | √ |
| 5. Solusi cryptarithmetic benar untuk persoalan cryptarihtmetic untuk lebih dari dua buah operand. | √ | |