Tugas Kecil Strategi Algoritma

**Laporan Penyelesaian *Cryptarithmetic* dengan Algoritma Brute Force**

Oleh:

Kinantan Arya Bagaspati

13519044



PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

1. **Algoritma Brute Force**

Algoritma brute force singkatnya ialah cara penyelesaian masalah dengan program yang meninjau semua kasus yang mungkin muncul berdasarkan deskripsi masalah yang diberikan. Meskipun dengan definisi yang diberikan, tetap ada banyak cara berbeda dalam menyelesaikan satu persoalan yang sama dengan metode brute force. Keragaman ini disebabkan pendekatan yang berbeda dalam mengerjakan soal, beberapa observasi yang dapat mengoptimalkan dan membuang kasus yang tidak perlu, atau pemilihan aspek mana yang di-brute force-kan. Secara keseluruhan, brute force merupakan algoritma yang mudah ditemukan dan diimplementasikan, namun membutuhkan waktu dan memori yang banyak.

Dalam menyelesaikan tugas kecil cryptarithmetic yang diberikan, saya menggunakan brute force yang mengutamakan kecepatan program. Terdapat dua hal utama yang berperan pokok dalam optimasi waktu program saya. Yang pertama ialah program ini melakukan prekomputasi sebanyak mungkin agar tiap pengecekan kasus bisa sesingkat mungkin. Kemudian program hanya meninjau permutasi kemungkinan pemetaan huruf yang ada ke 10 kemungkinan digit, bukannya mengecek semua kemungkinan digit per hurufnya. Terakhir, program tidak meninjau kasus permutasi yang sudah dicek atau melanggar aturan solusi cryptarithmetic. Berikut langkah yang dilakukan program saya secara urut.

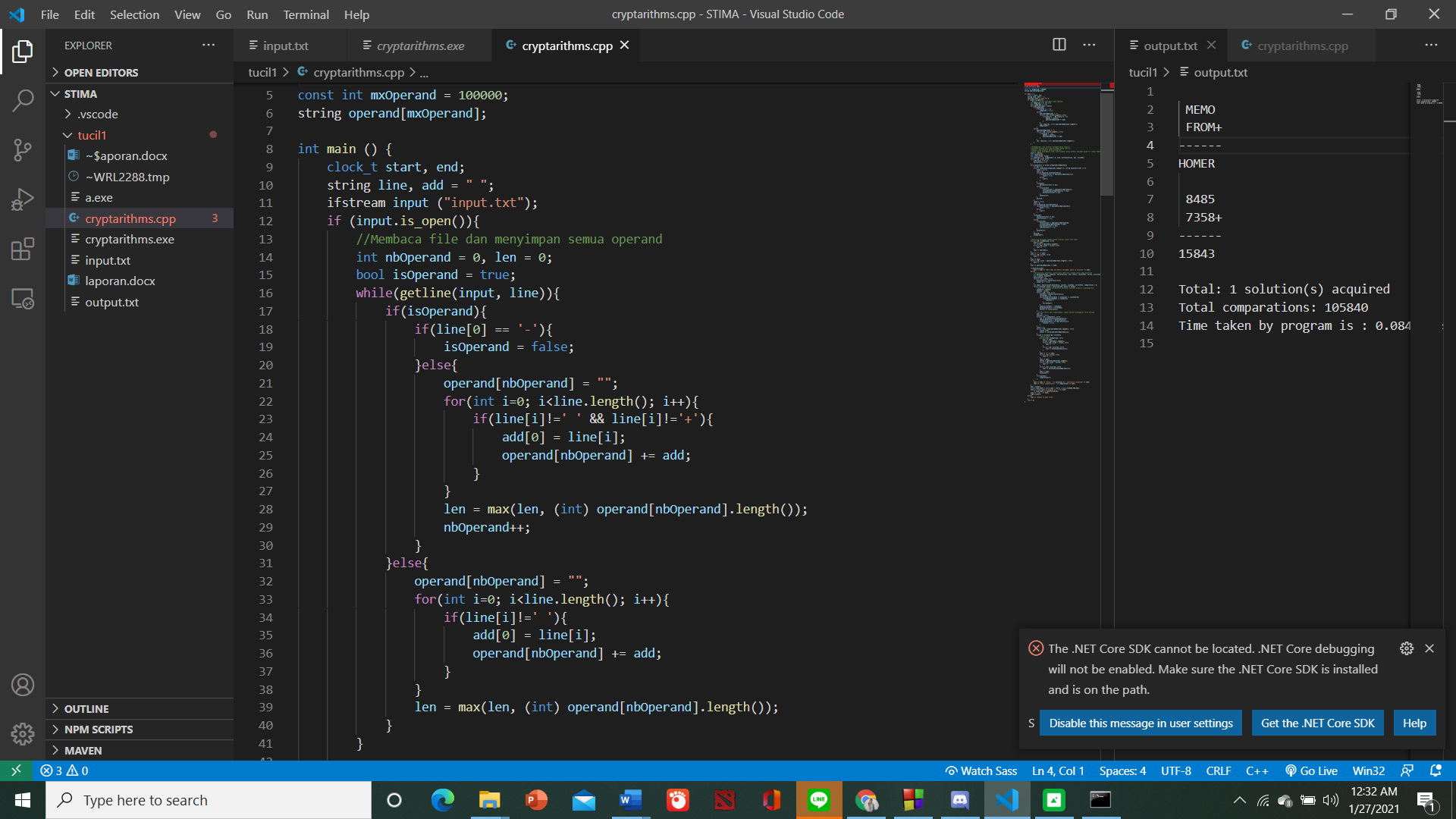
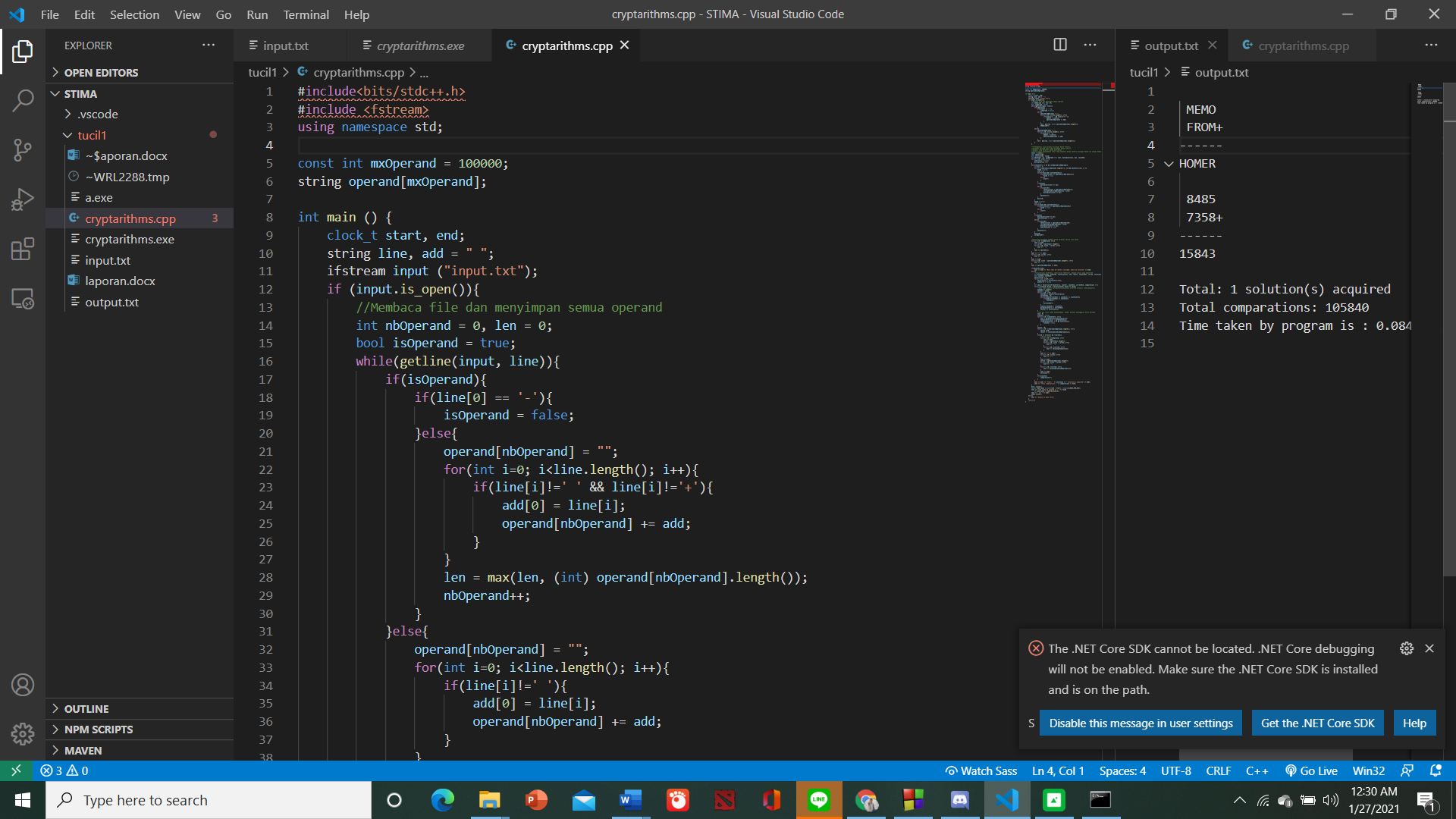
Pertama, program akan membaca file input baris demi baris. Tiap baris akan dibersihkan karakter ‘ ‘ dan ‘+’ dan akan disimpan kata demi kata dalam array of string operand[] dengan variabel nbOperand dan len masing-masing menyatakan banyak operand terbaca dan panjang kata terpanjang. Pembacaan terus berlangsung hingga terbaca baris dengan karakter pertama ‘-‘, kemudian kata berikutnya akan disimpan dalam operand[nbOperand].

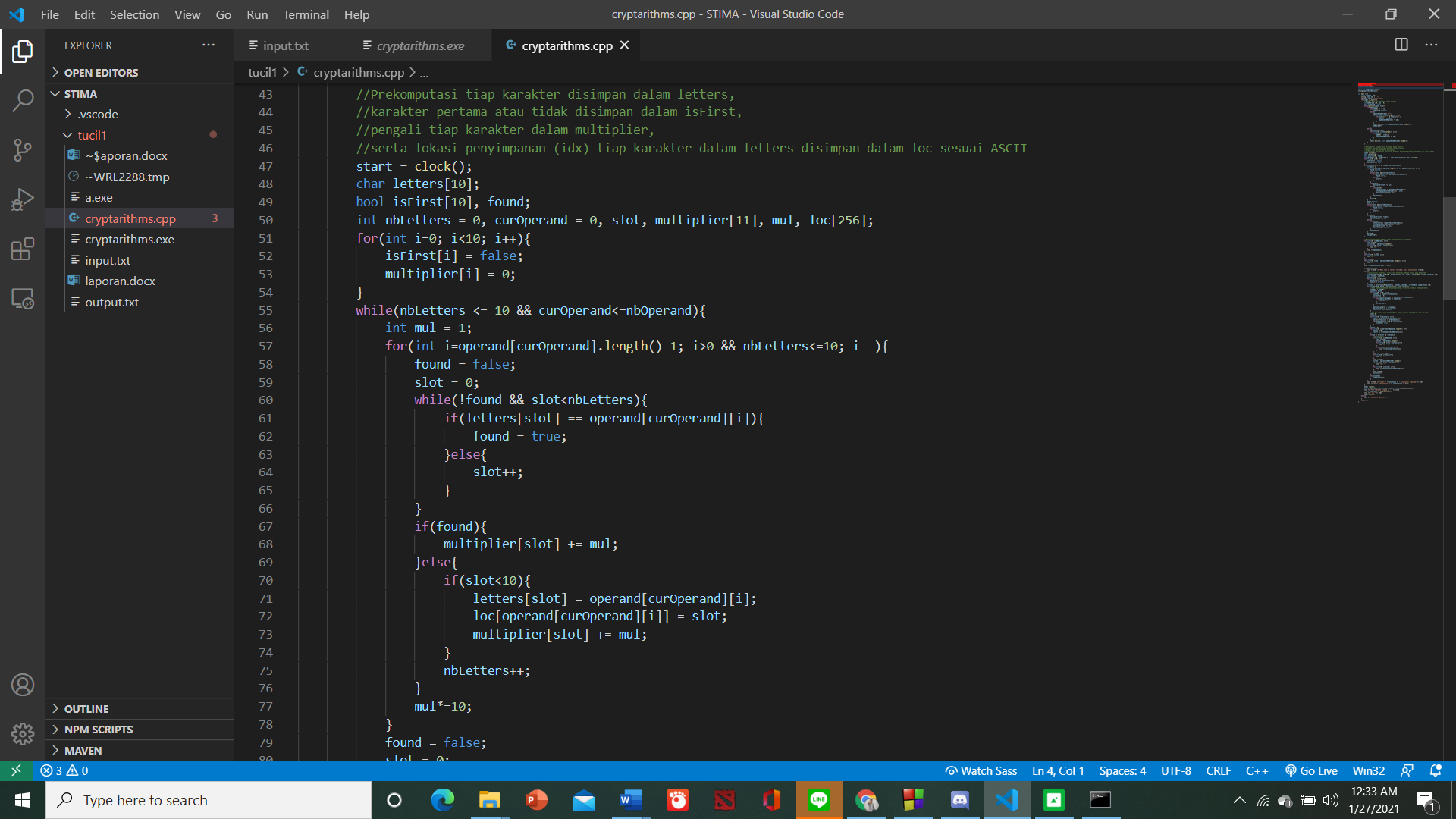
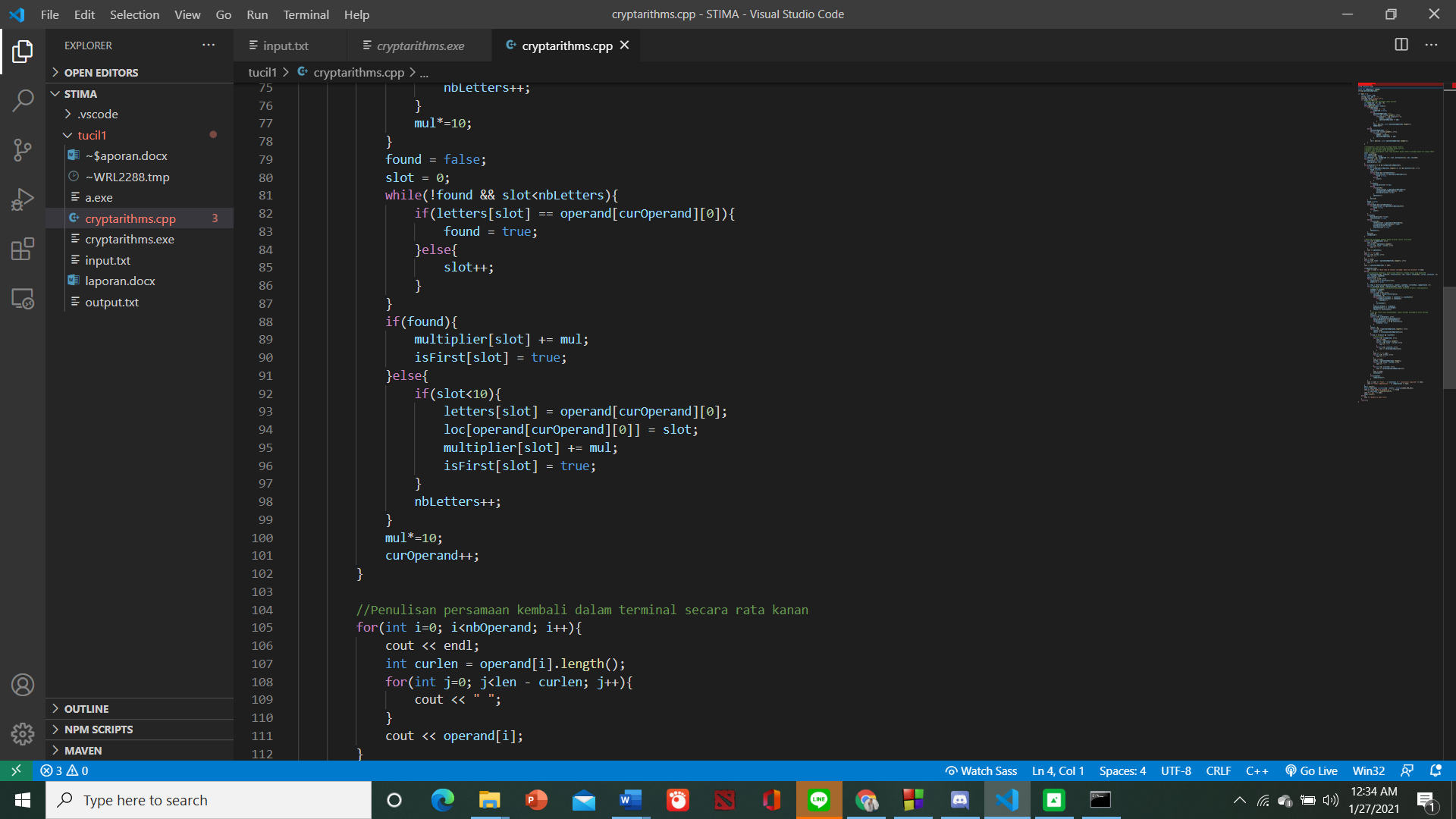
Kemudian disinilah perkomputasi dilakukan. Setiap string operand akan ditinjau karakter per karakter sehingga tiap karakter unik yang ditemukan disimpan dalam array letters. Array of boolean isFirst menyimpan apakah karakter ke-i dalam array menjadi karakter pertama dalam suatu kata atau tidak. Array multiplier menyimpan pengali tiap karakter, dan digunakan saat perbandingan dalam kasus2 yang ditinjau nanti. Terakhir, lokasi/indeks tiap karakter tersimpan dalam letters disimpan dalam array loc sesuai ASCII dari karakter tersebut.

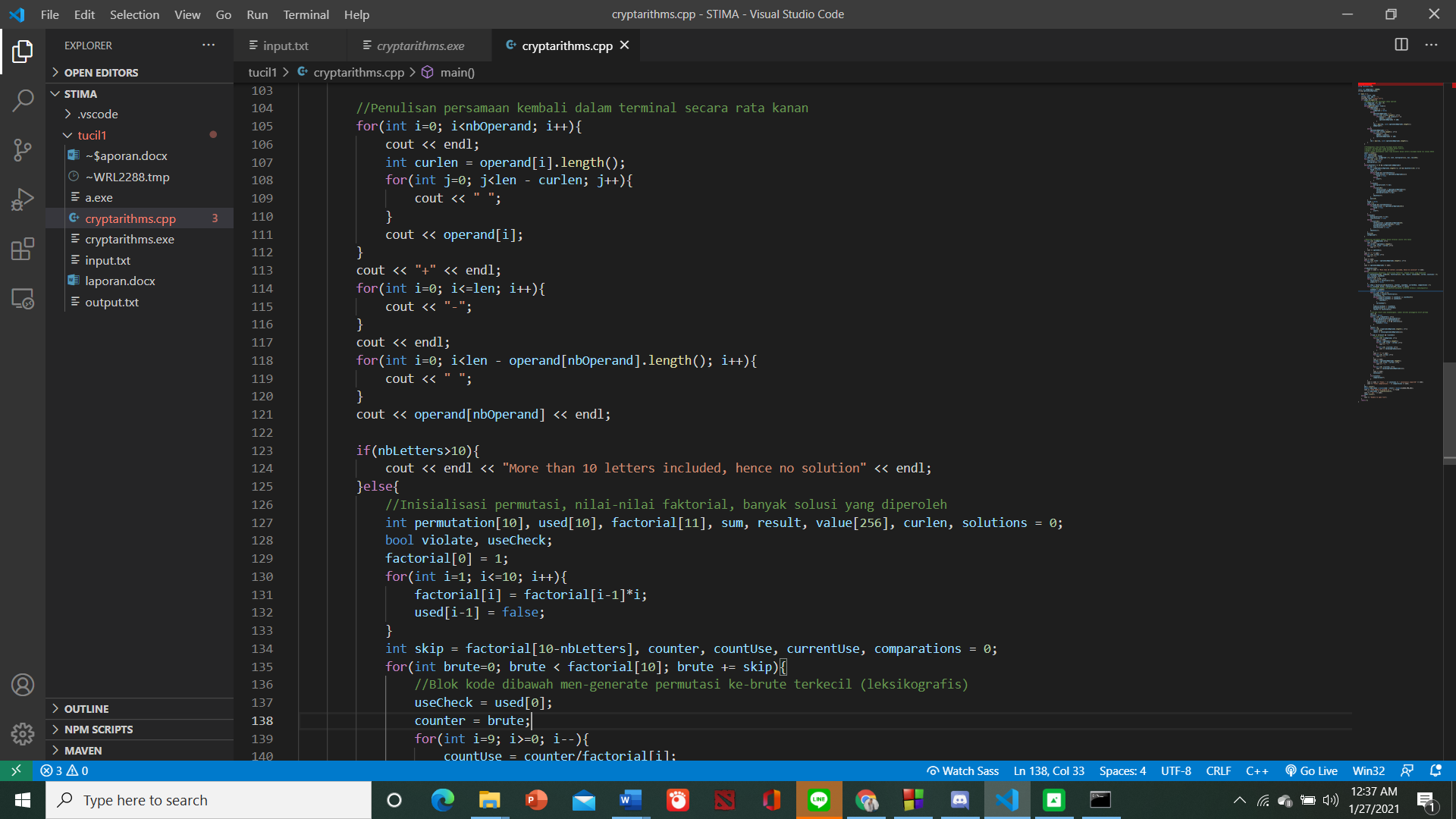
Kemudian dimulailah proses bruteforce dengan semua optimasi yang ada. Pertama program menggunakan array bernama permutasi untuk mensubstitusikan nilai ke tiap karakter dalam letters. Namun program tidak meninjau semua permutasi bila karakter yang lebih kecil dari 10, melainkan program menghitung variabel skip yakni (10-nbKarakter)!, sehingga brute force dilakukan pada permutasi ke - 0, skip, 2\*skip, dan seterusnya. Oleh karena itu, terdapat blok kode di awal setiap kasus yang menerima bilangan x dan menghasilkan sebuah permutasi ke-x terkecil secara leksikografis.

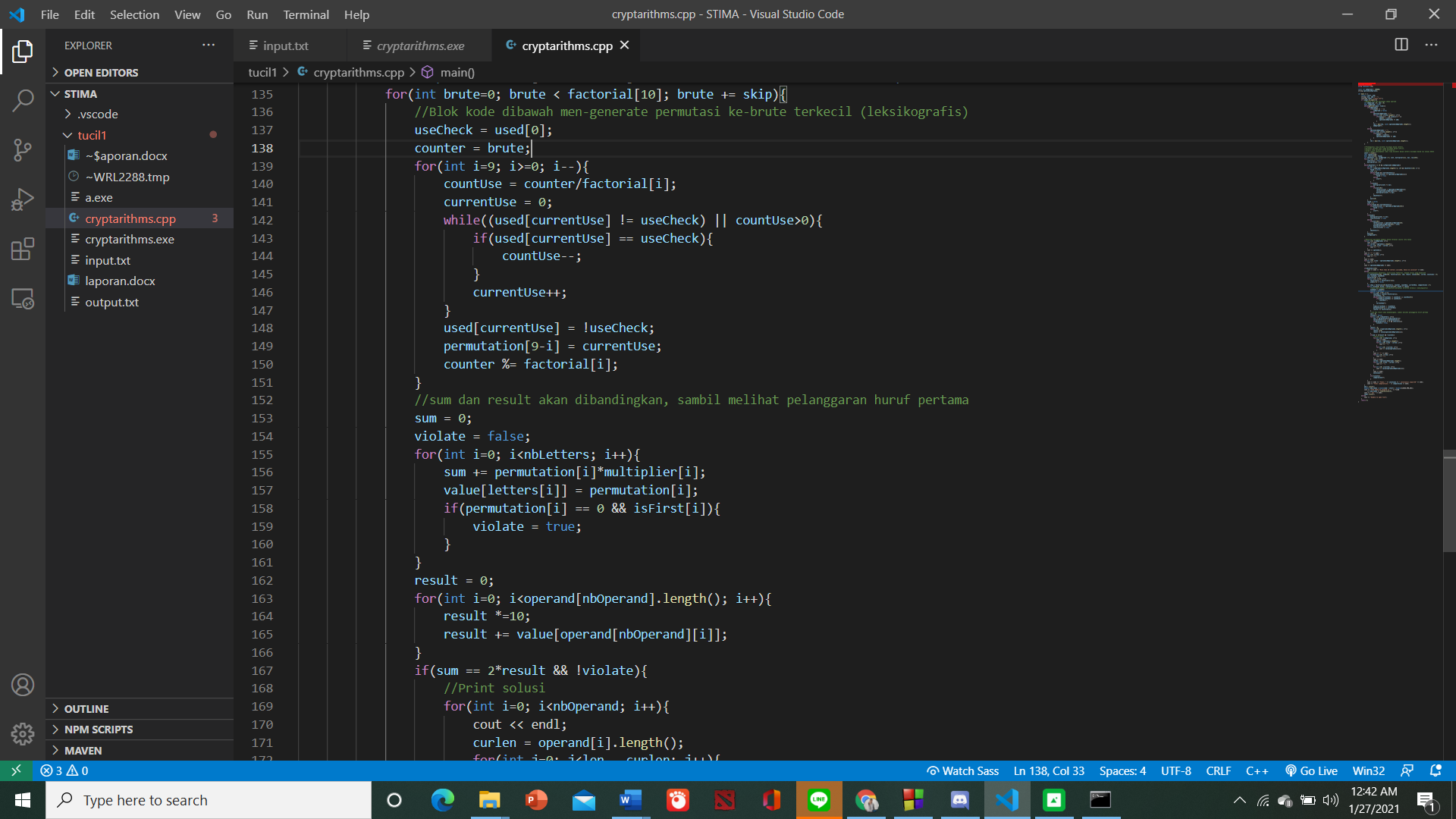
Permutasi ini kemudian disubstitusikan ke masing-masing karakter, dicek apakah tidak ada yang melanggar huruf pertama bernilai 0, serta dibandingkan jumlah semua operand dengan hasil menggunakan pengali yang sudah diprekomputasi. Apabila diperoleh kecocokan, hasil akan langsung diprint sesuai format yang ada. Kemudian di akhhir, program akan menampilkan banyak solusi yang diperoleh, pencocokan yang terjadi, serta waktu yang dibutuhkan.

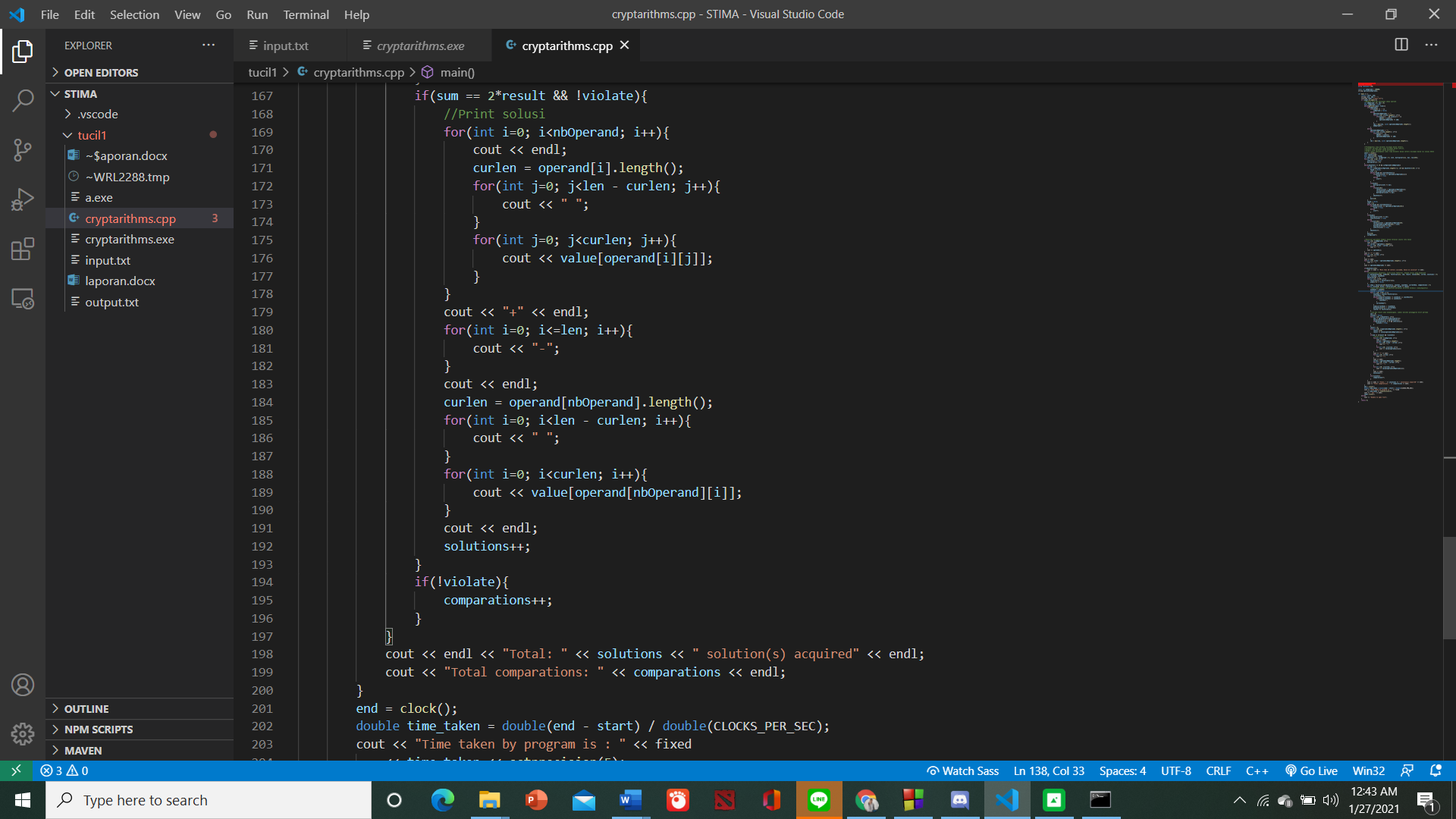
1. **Source Program**

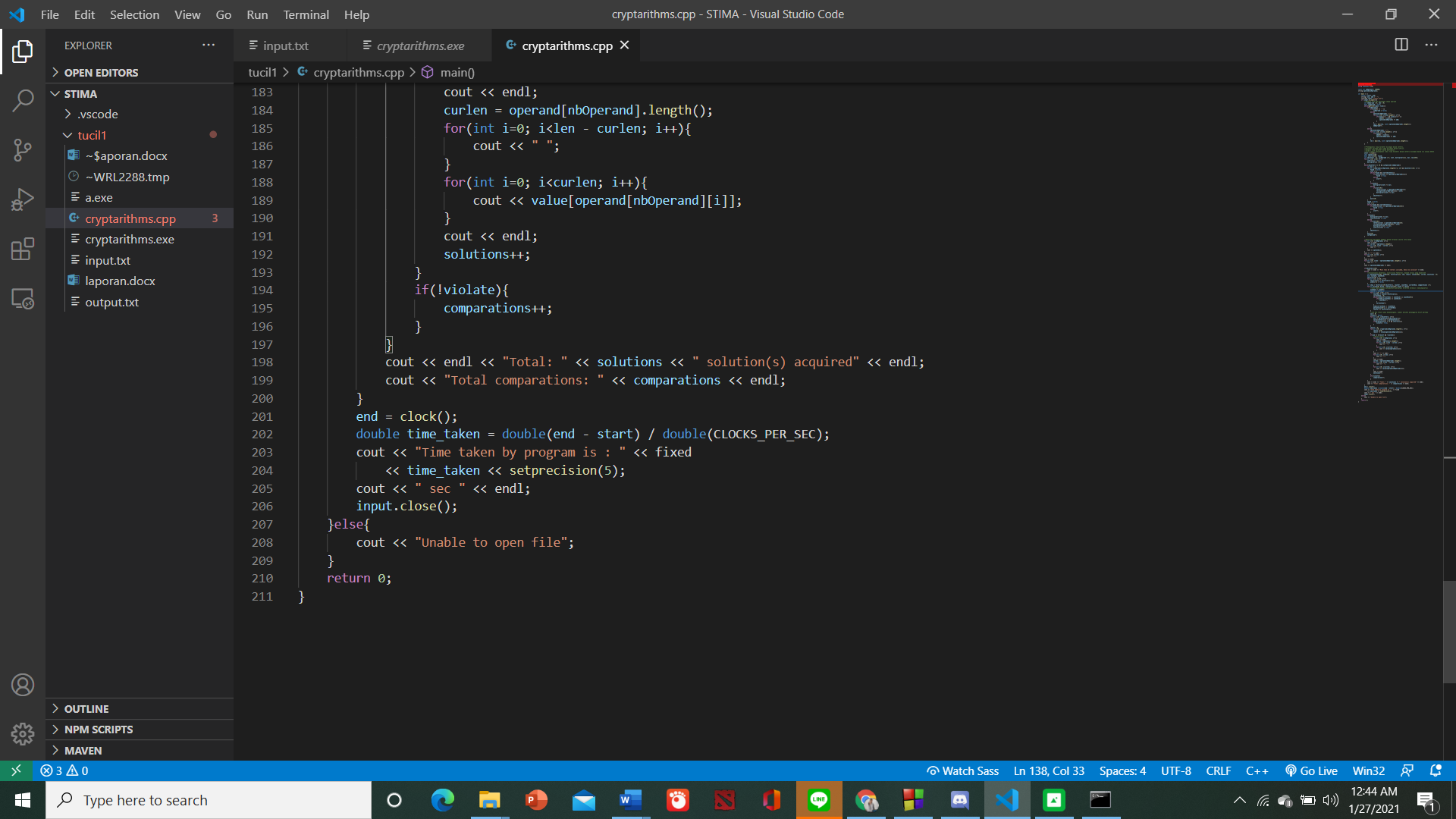


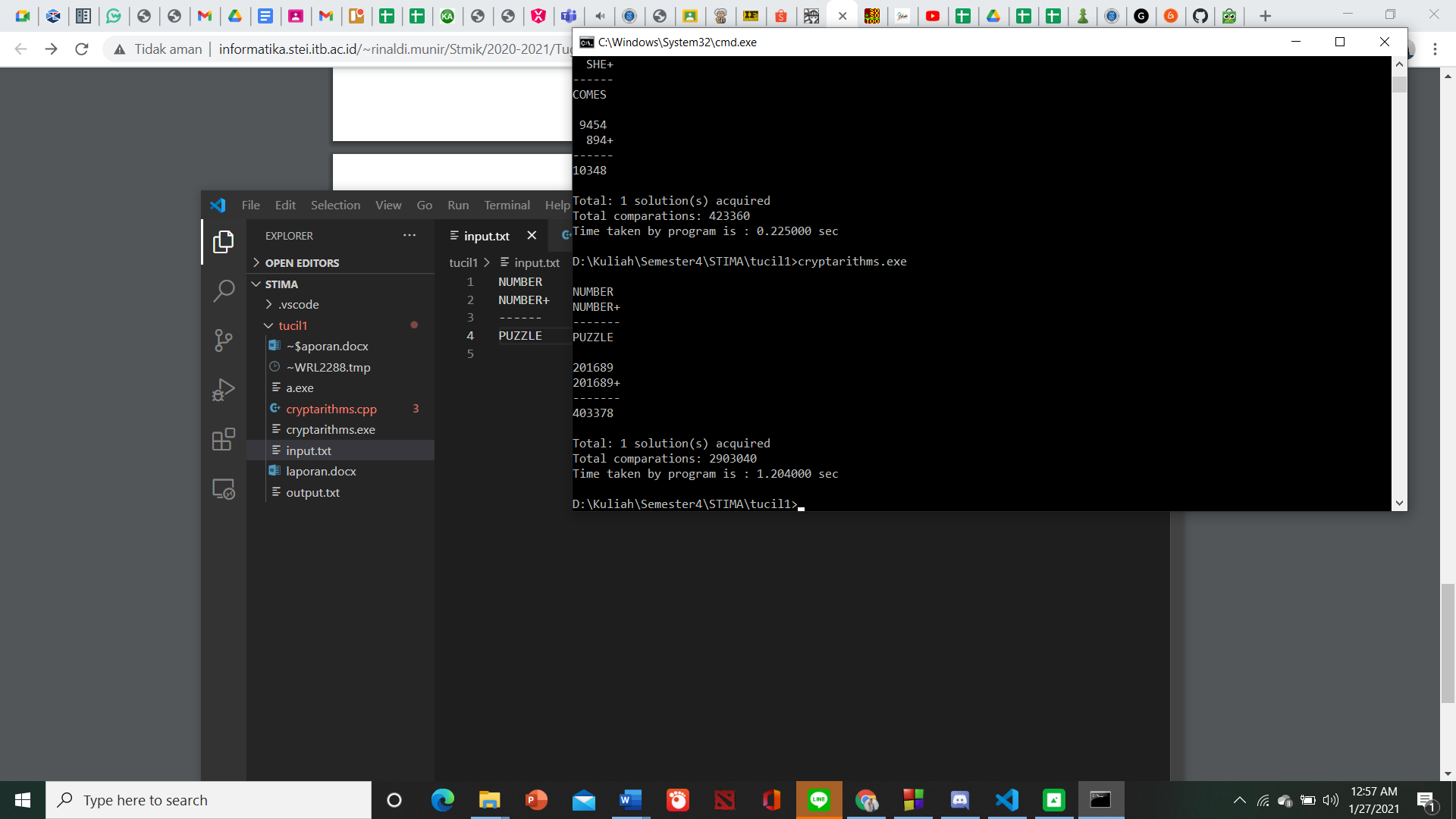
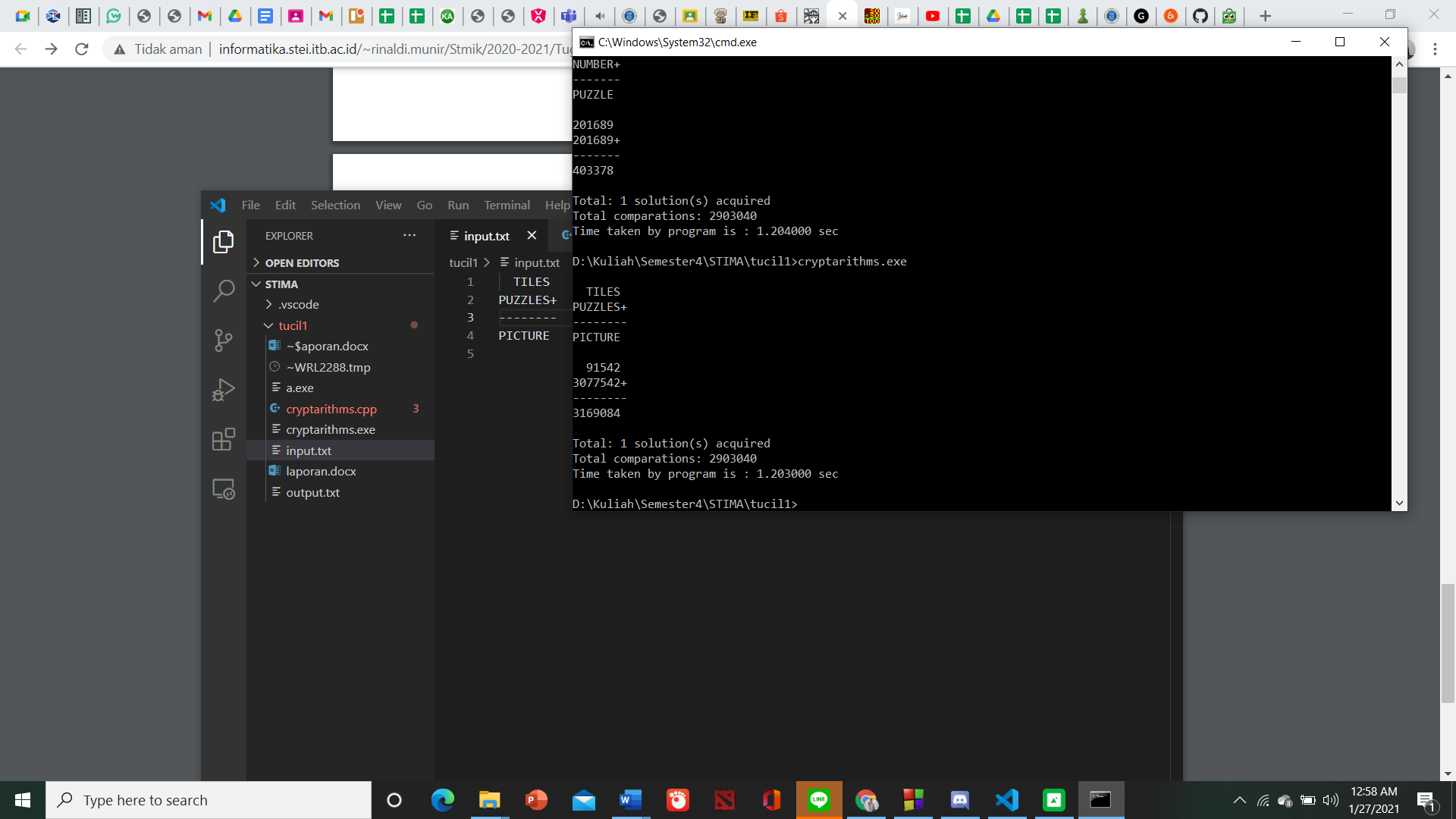
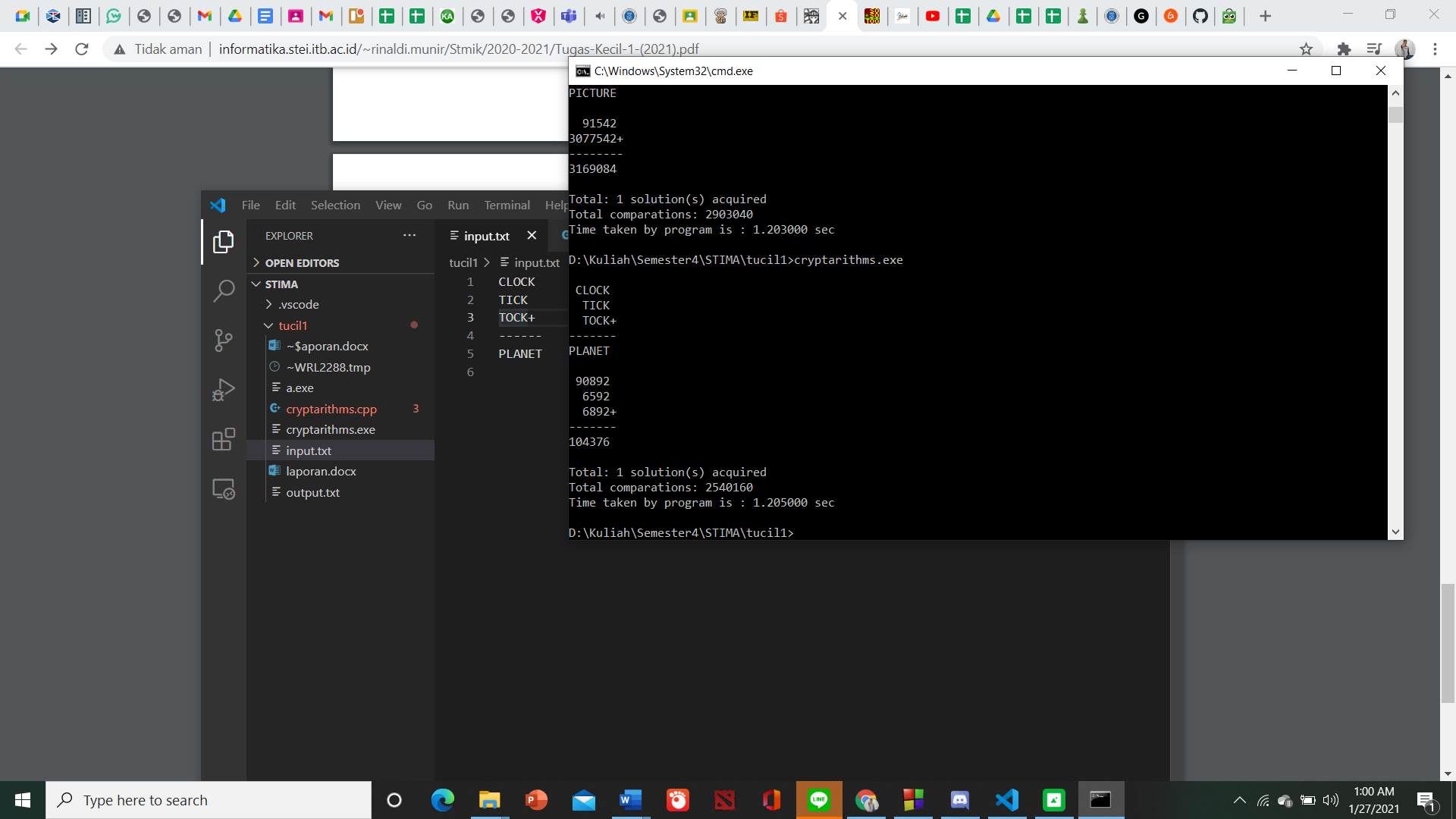
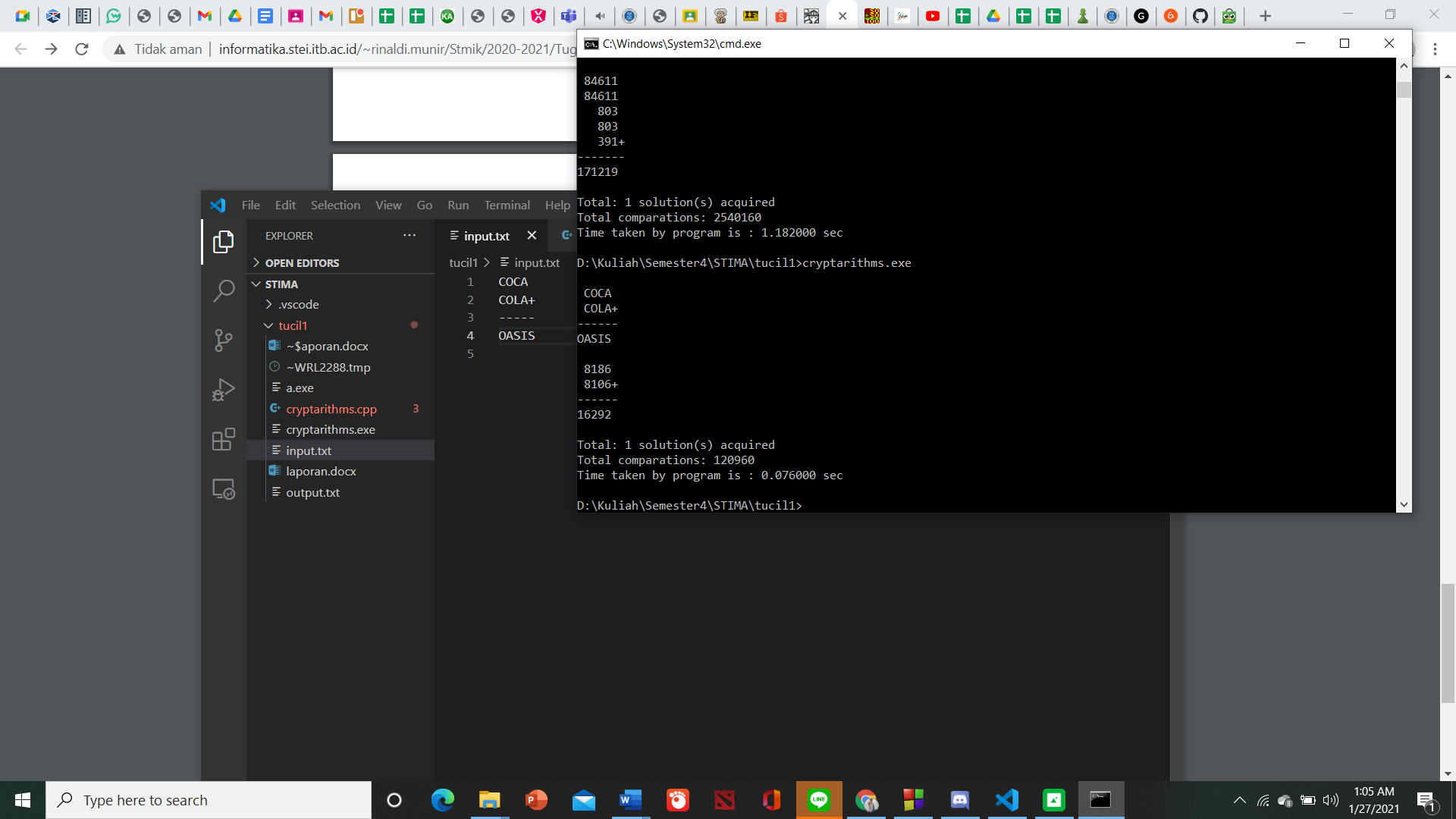
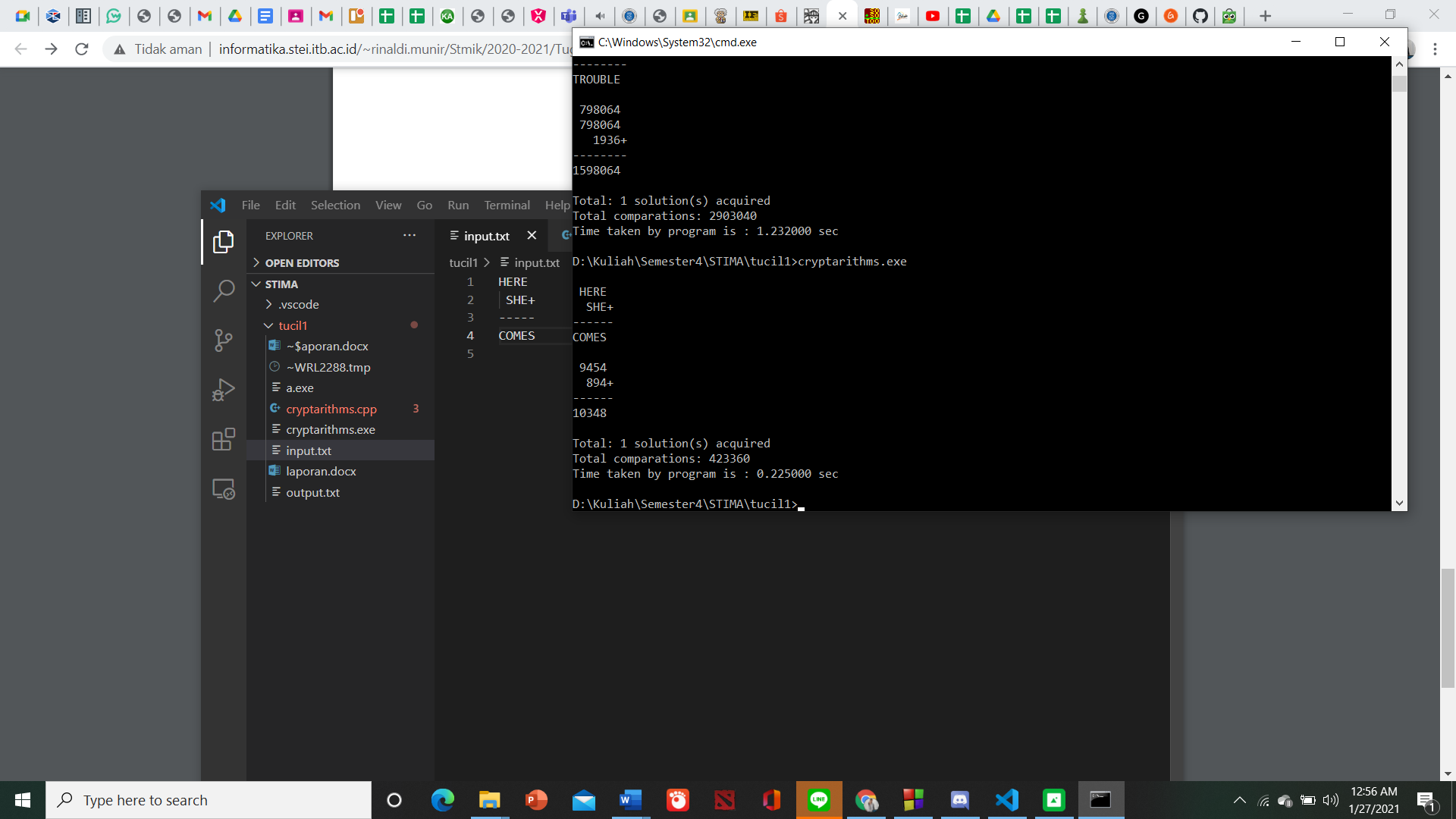
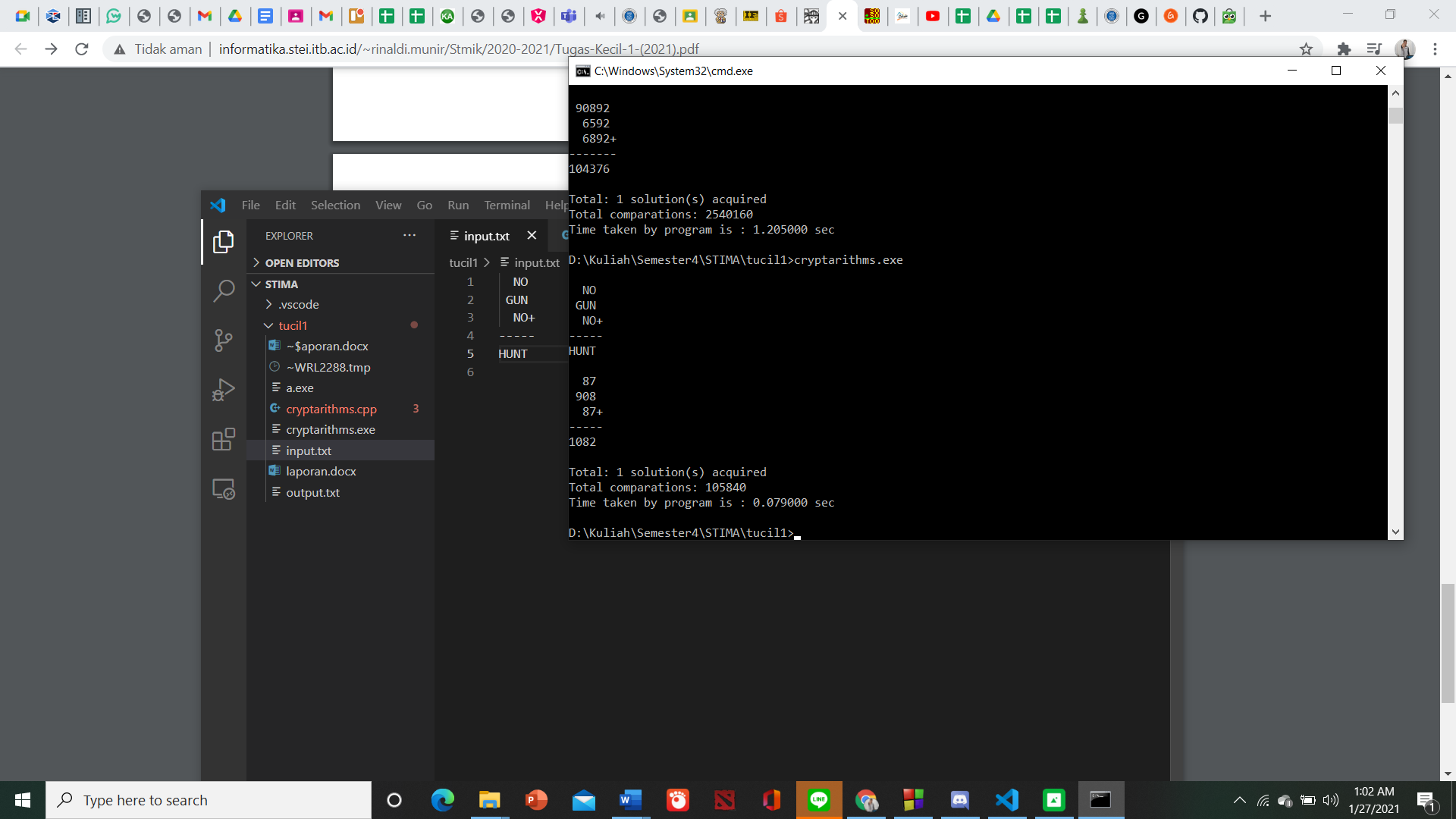
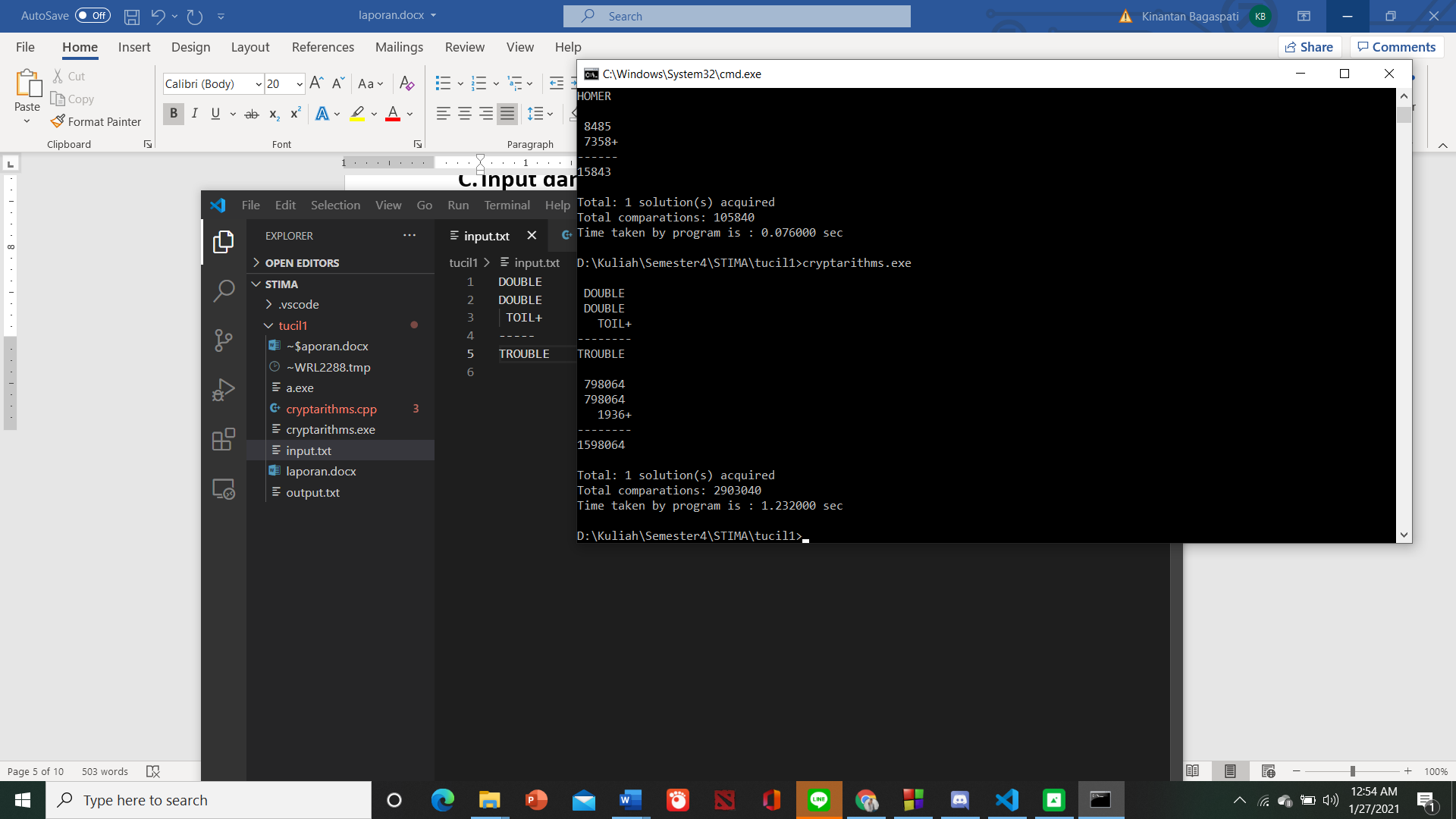


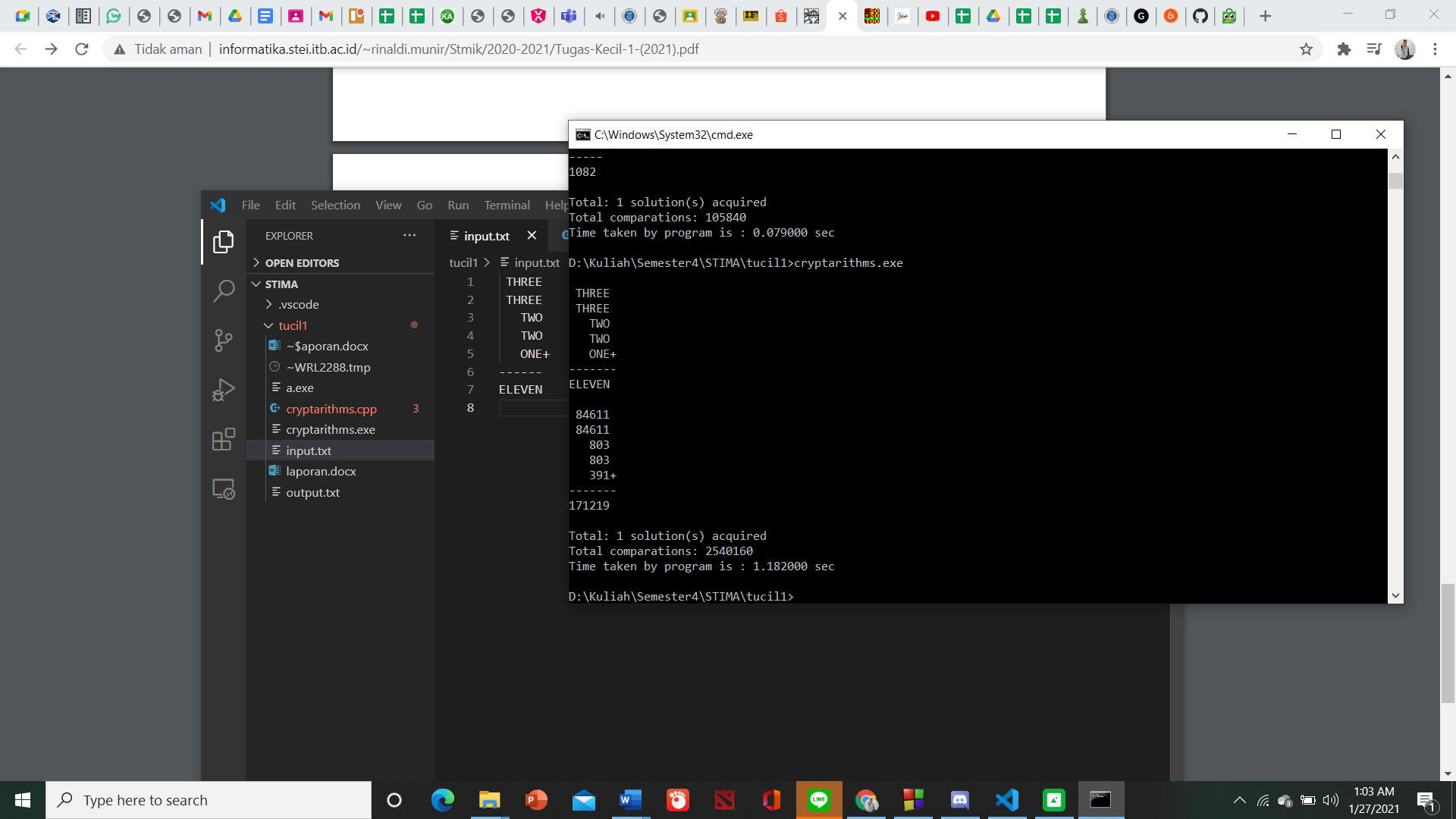
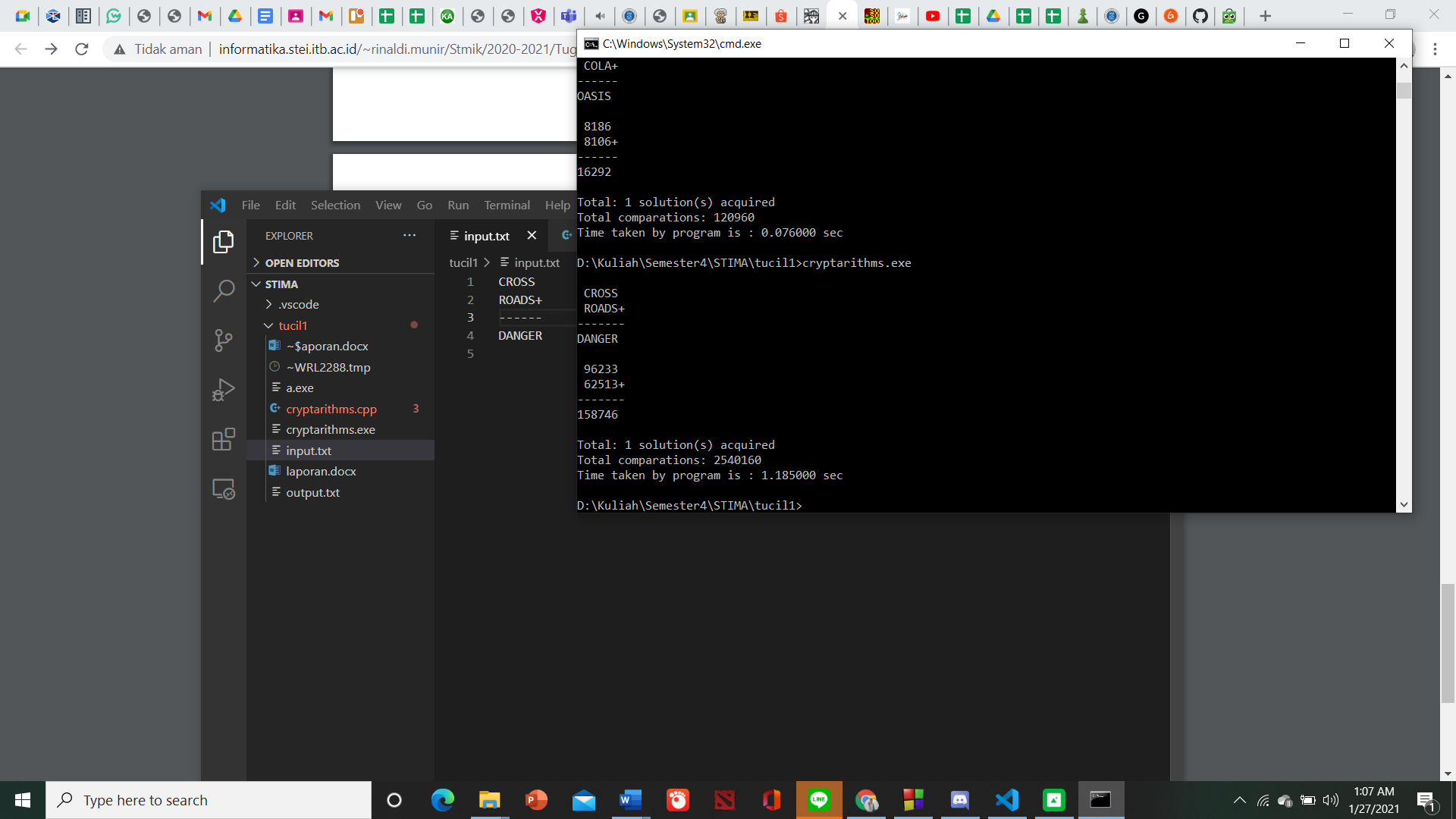
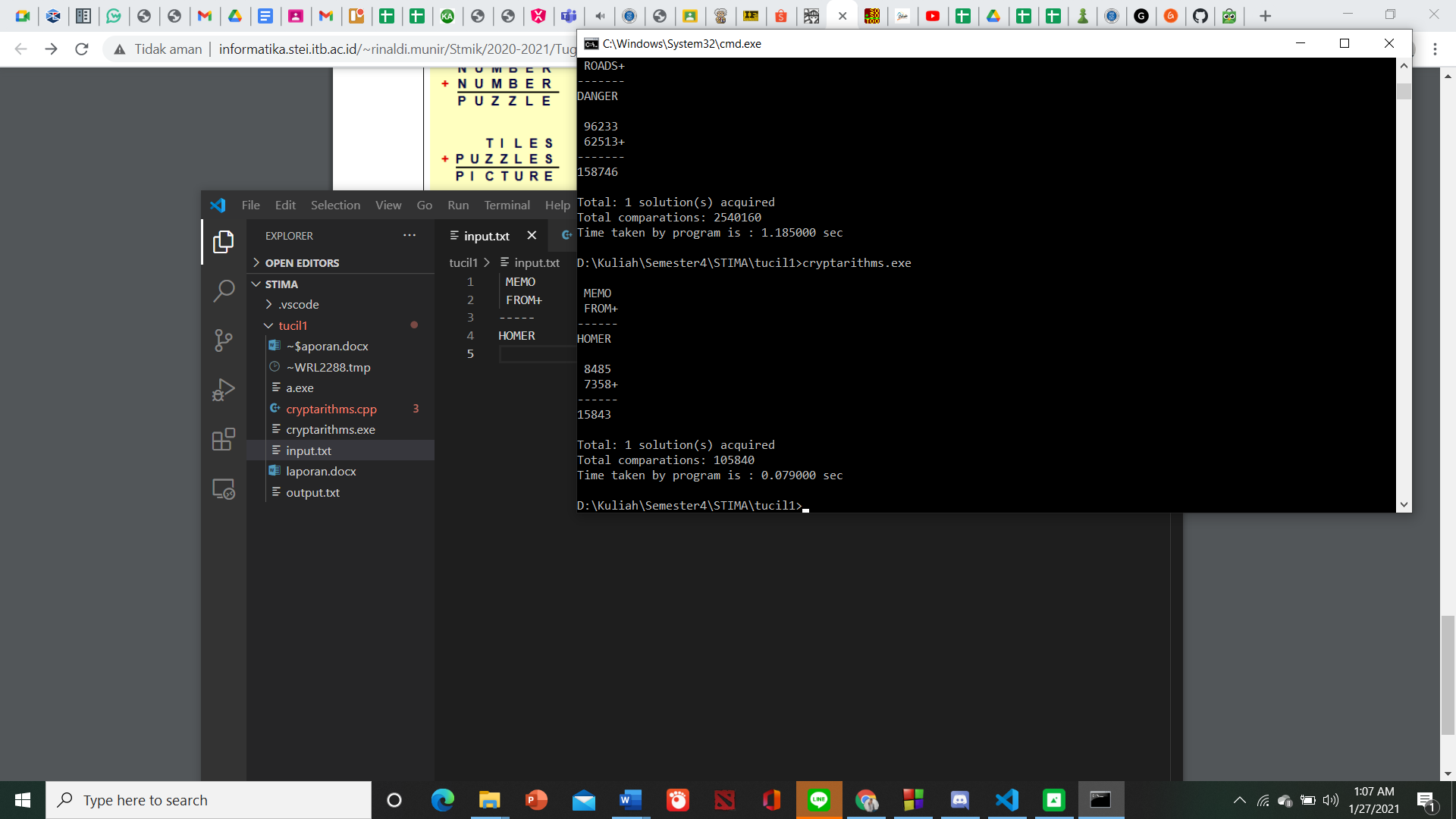






1. **Input dan Output**

1. **Alamat Drive**

<https://drive.google.com/drive/u/1/folders/1j6dxA34jl_ooia1Cx7bE-1LFLIj6dMq_>

1. **Tabel Ceklist**

|  |  |  |
| --- | --- | --- |
| **Poin** | **Ya** | **Tidak** |
| Program berhasil dikompilasi tanpa kesalahan | √ |  |
| Program berhasil running | √ |  |
| Program dapat membaca file masukan dan menuliskan luaran | √ |  |
| Solusi cryptarithmetic hanya benar untuk persoalan cryptarithmetic dengan dua buah operand. |  | √ |
| Solusi cryptarithmetic benar untuk persoalan cryptarihtmetic untuk lebih dari dua buah operand. | √ |  |