**NAME: SHUBHAM SHARMA**

**ROLL NO: 18i190002**

**MSC PHD (OR)**

**PART (1) and PART(2)**

Given that

A={a1,a2,a3,…,an} is the set of symbols that we’ll have in the string

A string s=s1 s2 s3… sl of length l where each character belongs to A

Given S={s1, s2,…, sM}

Aim is to find the string c=c1 c2 c3 c4…. cl such that is minimum.

We can do this by giving each alphabet in A a number(eg 1 , 2 , 3 , 4)

Defining a parameter l where l is the length of each string.

Defining a set STRCH={1,2,3,…,l}.

We give each ai in A a number and let that set of numbers be NUM={1,2,3,…,n}

Defining a matrix named ‘string’ such that stringS\*STRCH where each element of the matrix will tell the corresponding number of that charcter of the string , eg strings1,3 will tell the number corresponding to the 3rd character of s1.

An example of ‘string’ can be:

**matrix** string:

1 2 3 4 5 6 7 8 9 10:=

s1 2 3 1 1 1 4 4 1 1 4

s2 2 3 3 4 4 4 2 1 4 1

s3 2 2 3 1 2 2 3 2 2 1

s4 2 1 1 4 2 3 2 3 2 1;

Now , Defining the decision variables:

Let x{S,STRCH} be variables in [S,STRCH] such that x is binary.

Let c{STRCH} be variables in STRCH such that c is integer.

Defining the objective function:

**MINIMIZE**

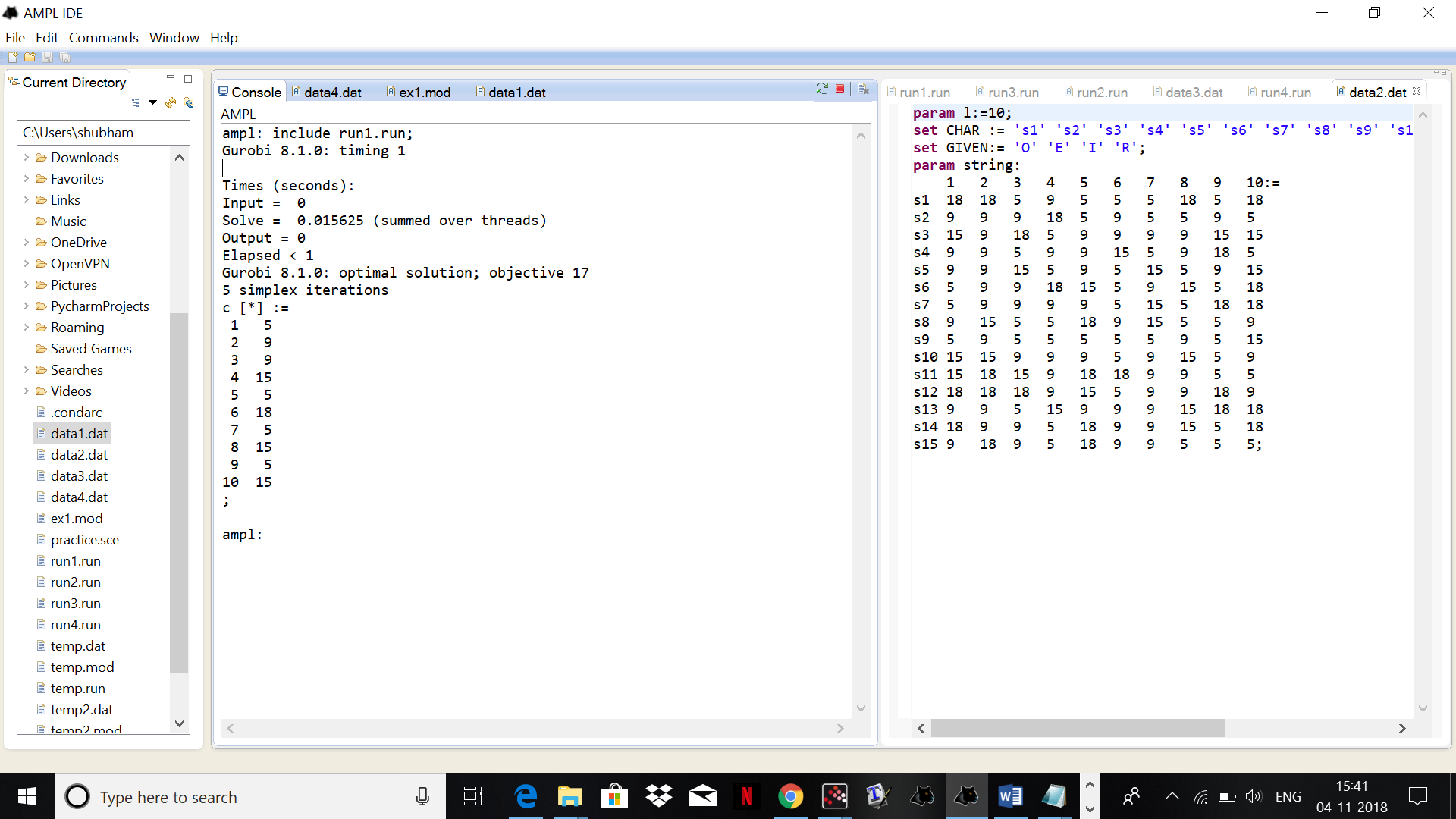
subject to constraints:-

x[i,j ] ≥ (c[j]-string[i,j])/50 , for all i in S, j in STRCH.

x[i,j ] ≥ -(c[j]-string[i,j])/50 , for all i in S, j in STRCH.

We have formulated the above question as an integer lenear programming problem.In the above problem, we have taken each character of the string to be a number and then made the constraints in such a way that it will give the make a string in such a way that it will minimize .

**PART (3)**



The string c is : ‘eiioereoeo’

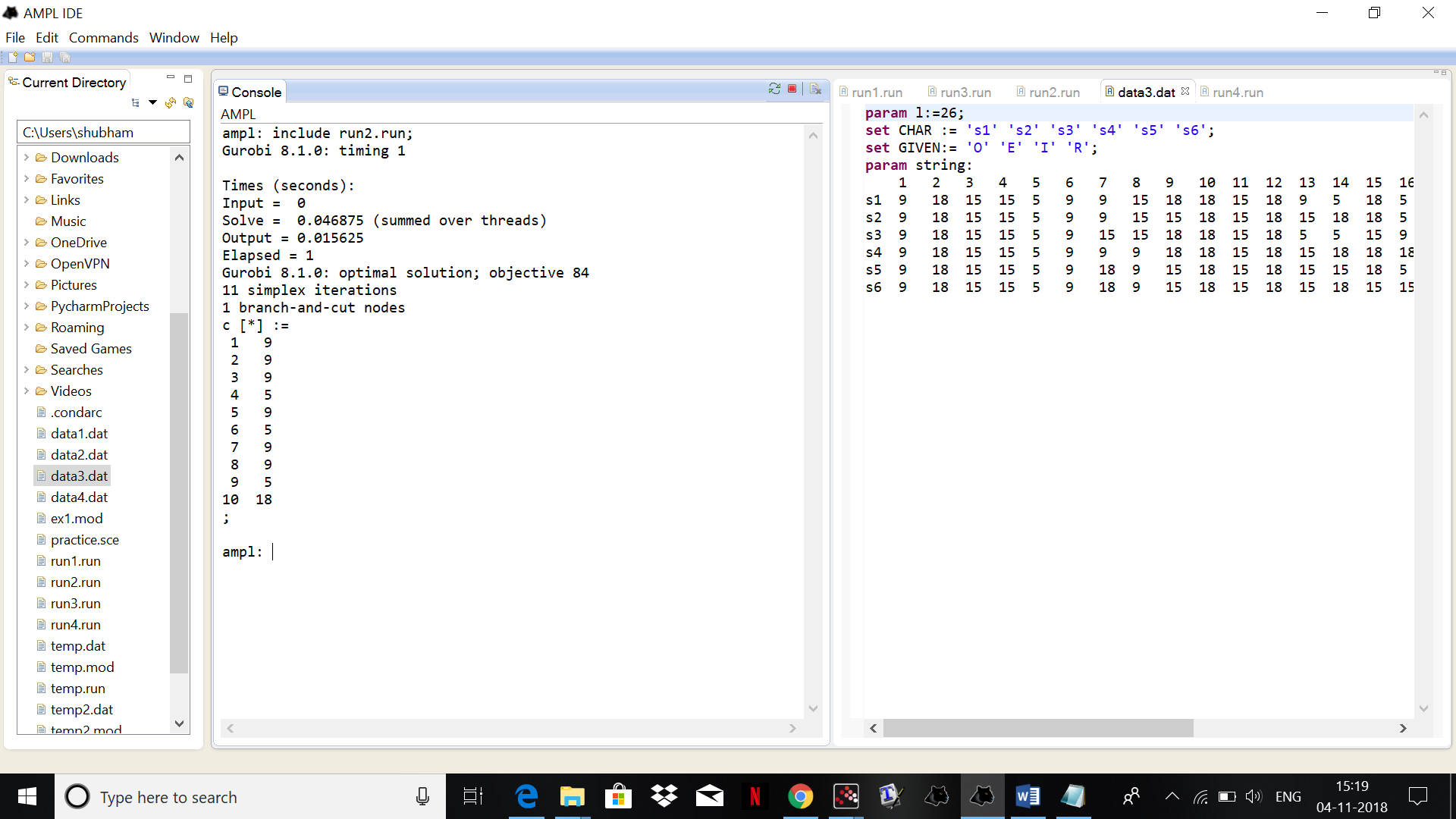
Times (seconds):

Input = 0.015625

Solve = 0 (summed over threads)

Output = 0

Elapsed < 1



The string c is :’iiieieiier’

Times (seconds):

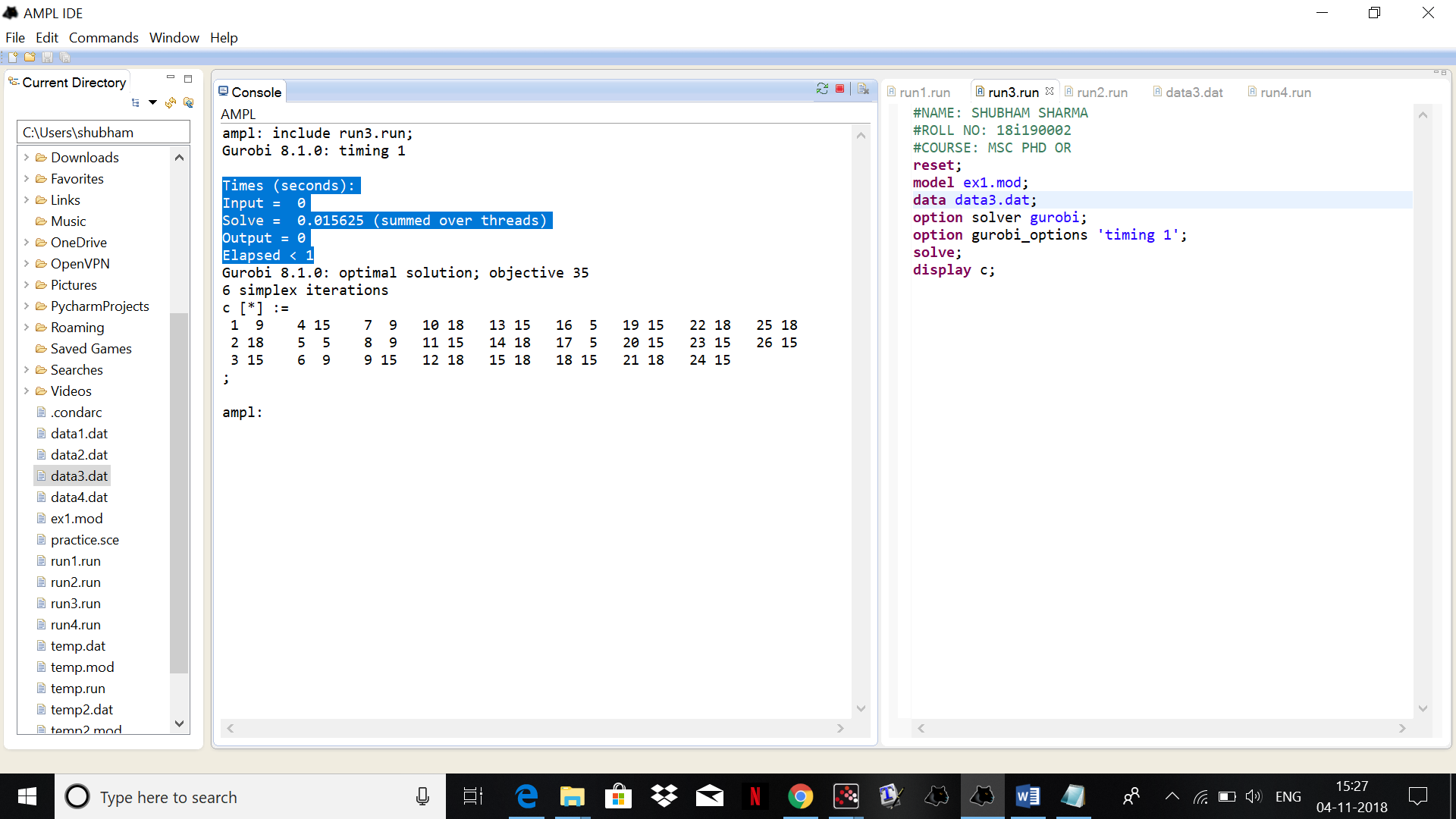
Input = 0

Solve = 0.046875 (summed over threads)

Output = 0.015625

Elapsed = 1

**PART (4)**



The string c is :’irooeiiiorororoeeooorrooro’

Times (seconds):

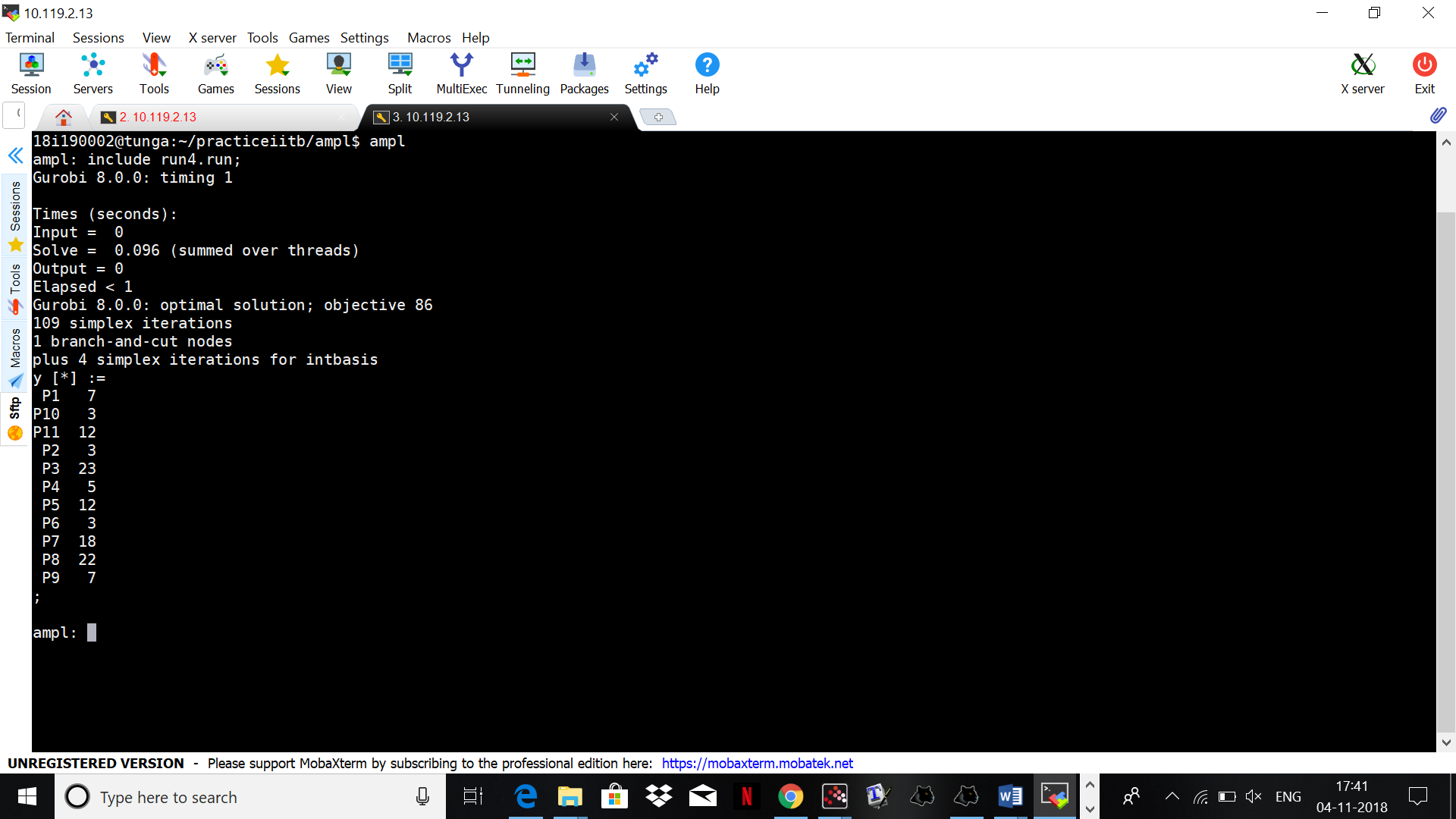
Input = 0

Solve = 0.015625 (summed over threads)

Output = 0

Elapsed < 1

**PART (5)**



the string c is : ‘gcwelcrvgcl’

Times (seconds):

Input = 0

Solve = 0.096 (summed over threads)

Output = 0

Elapsed < 1

**PART (6)**

For data 1

cohesiveness=2

For data 2

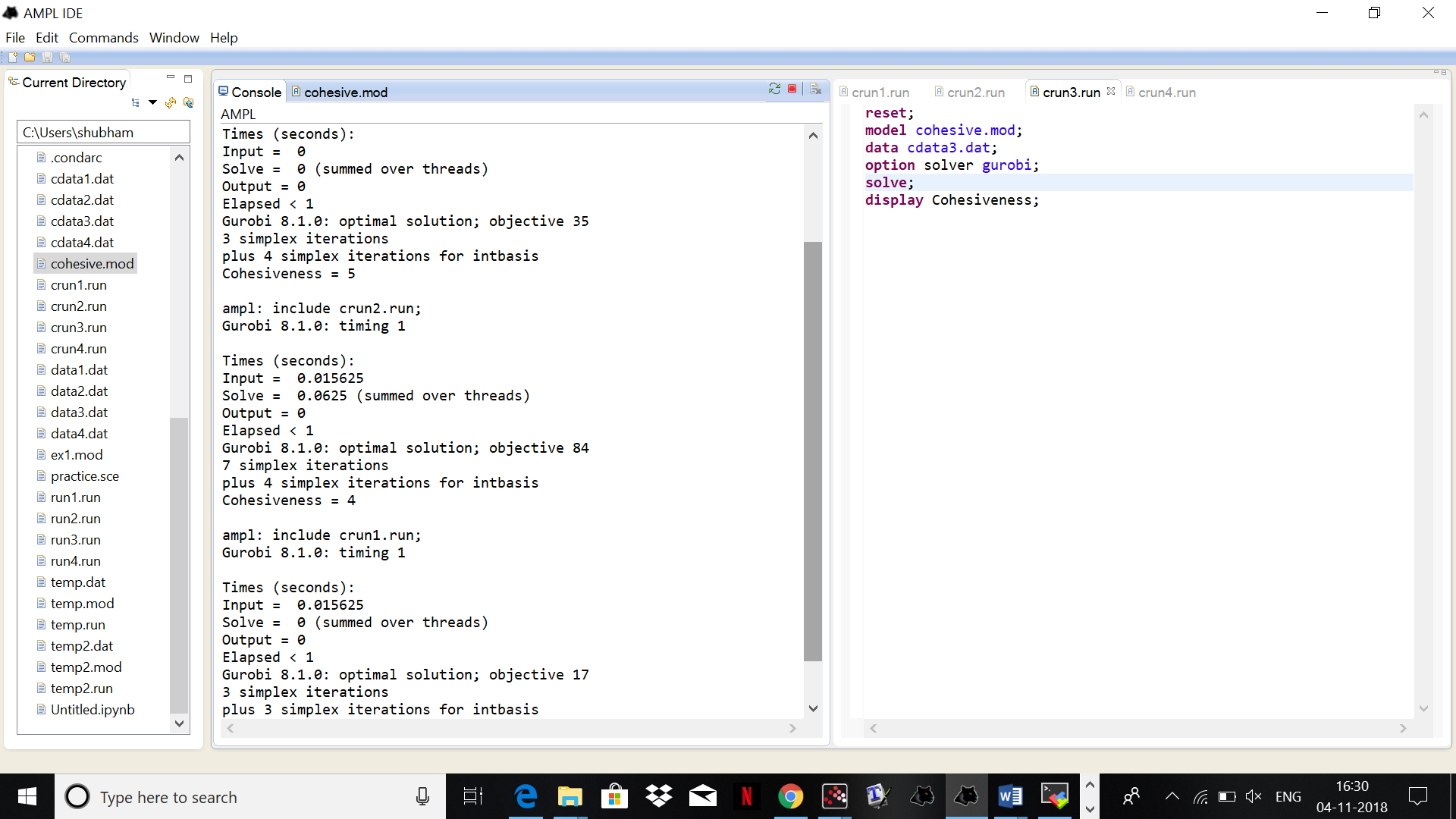
cohesiveness=4

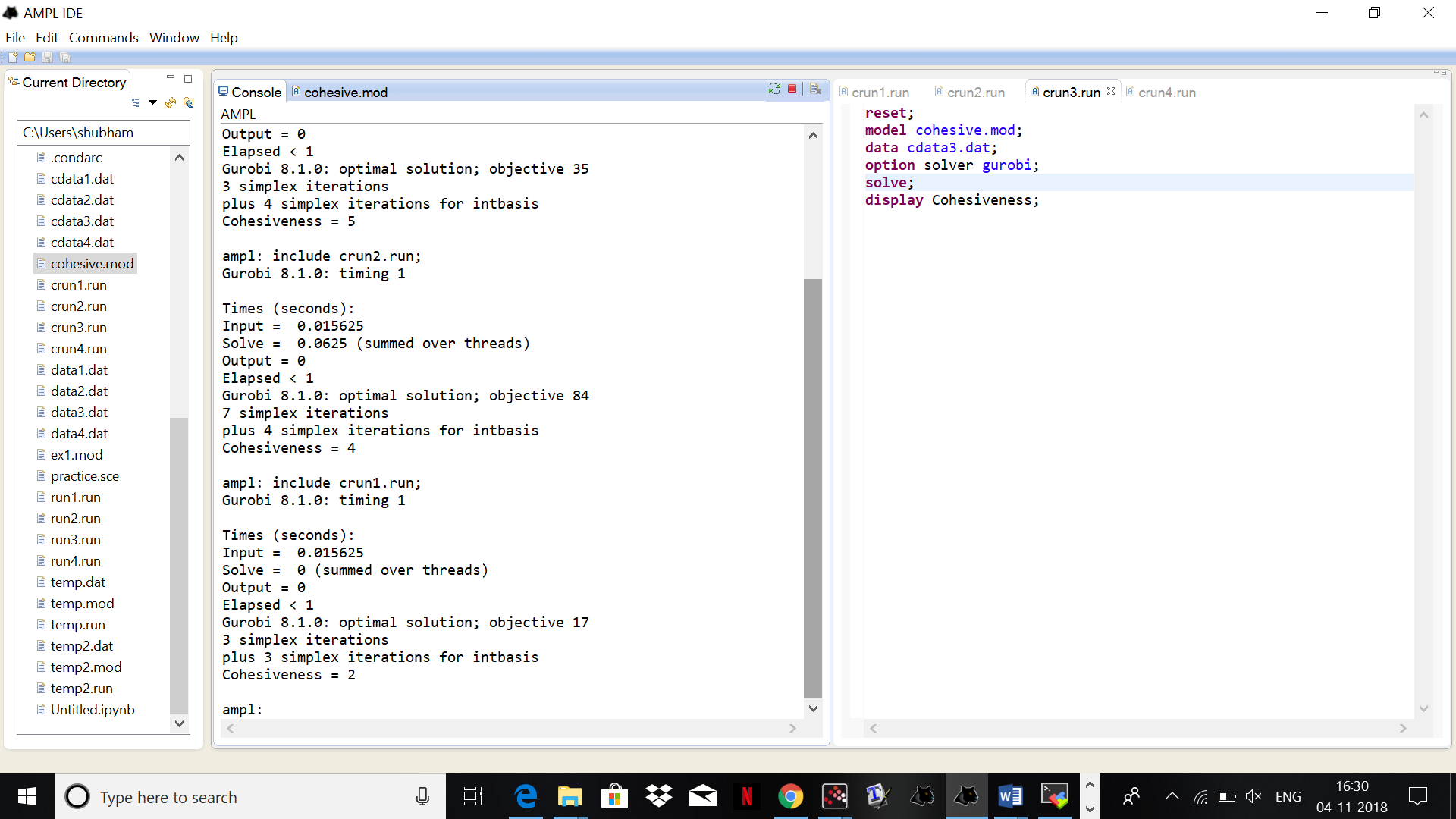
For data 3

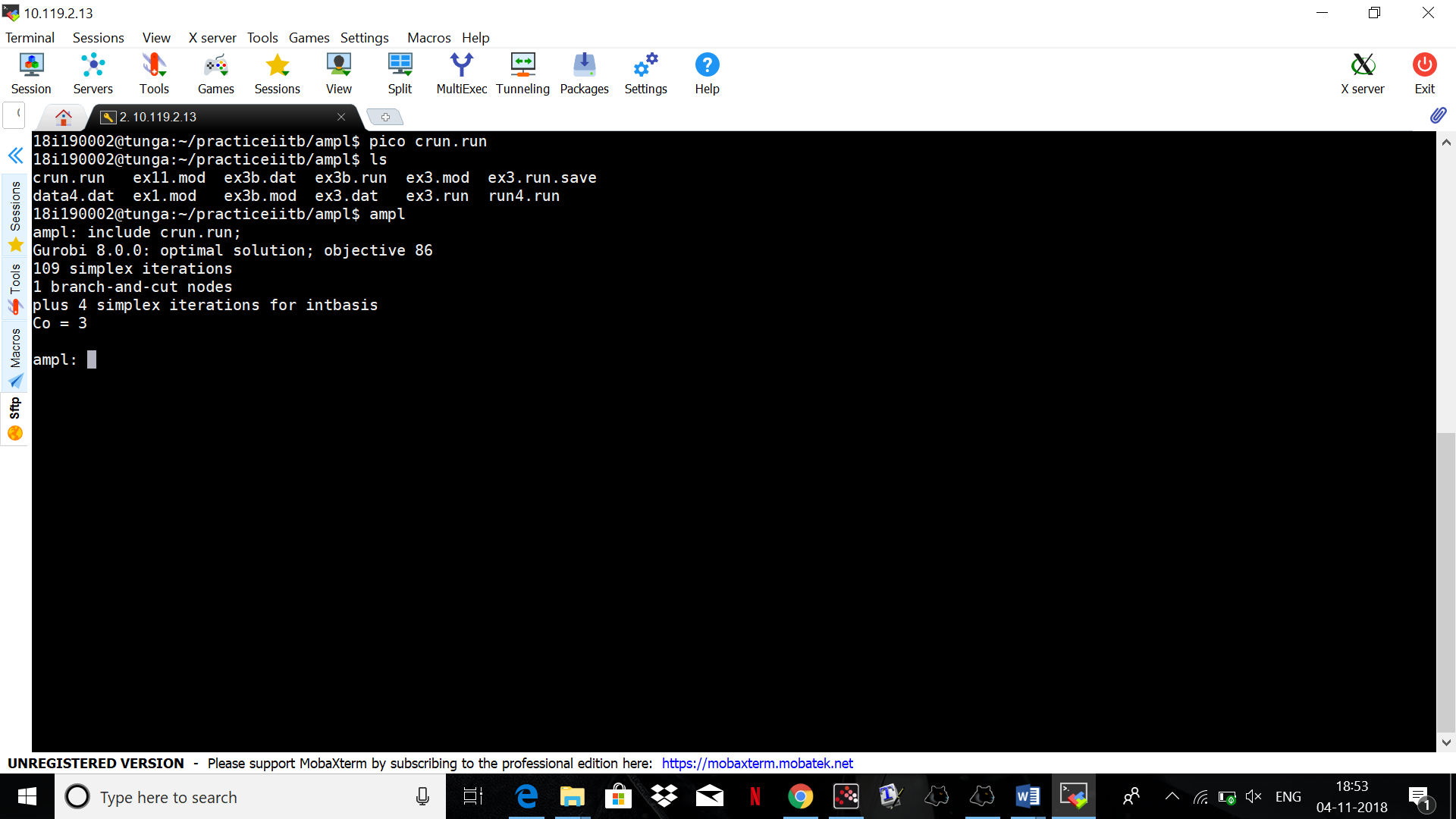
cohesiveness=5

For data 4

Cohessiveness=3







**PART (7)**

If the objective was to minimize Cohesiveness over all possible l length strings, then we would have made a optimisation model in such a way that it minimises the cohesiveness of all the l length strings that can be made by the given alphabets. We find the max and min of the sum of variables and find the cohesive minimum