RA1911033010114

UNNATI MISHRA

LAB EXPERIMENT 3:

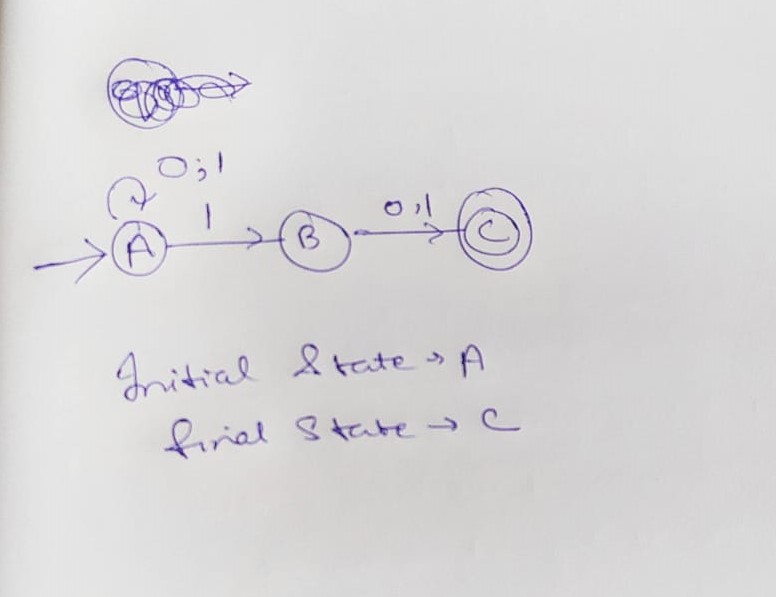
NFA TO DFA

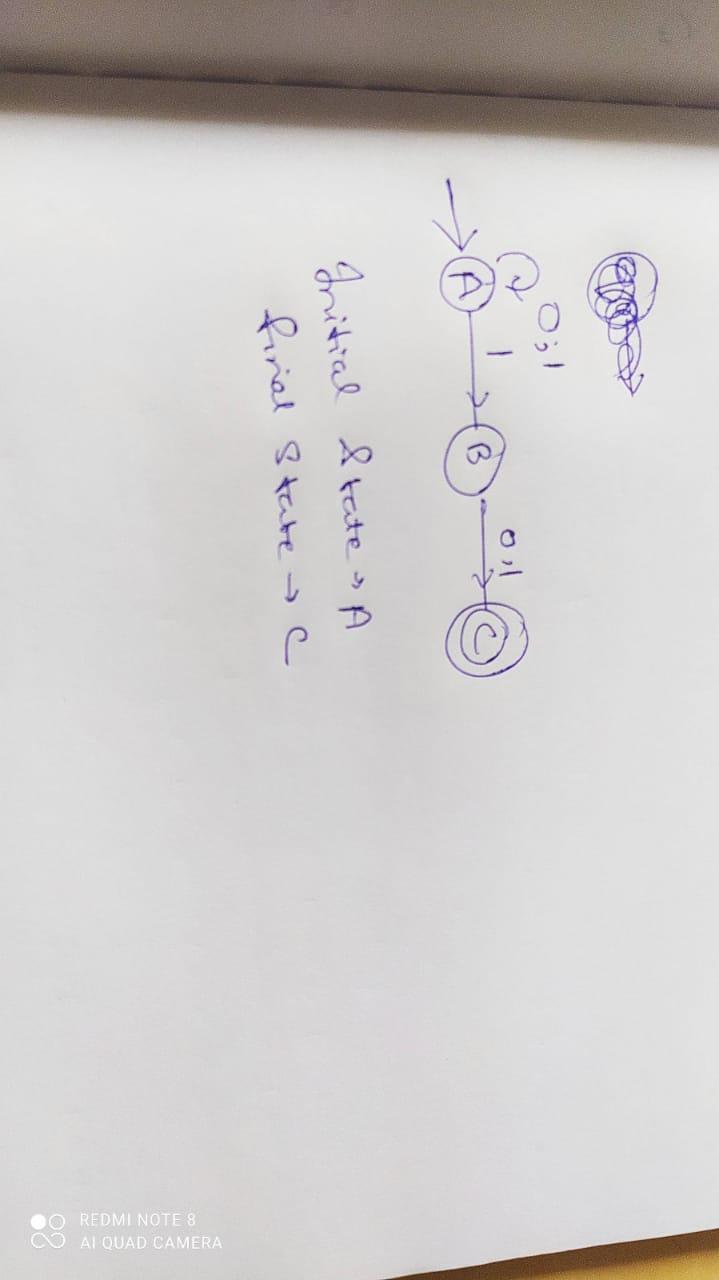
**AIM:** To write a program for converting NFA to DFA.

**ALGORITHM:**

1. Start
2. Get the input from the user
3. Set the only state in SDFA to “unmarked”.
4. While SDFA contains an unmarked state do:
5. Let T be that unmarked state
6. For each a in % do S = e-closure(move NFA(T,a))
7. If S is not in SDFA already then, add S to SDFA (as an “unmarked” state)
8. Set MoveDFA(T,a) to S
9. For each S in SDFA if any s and S is a final state in the NFA then, mark S and a final state in the DFA
10. Display the output
11. Stop

**INPUT:**

****



**PROGRAM:**

import pandas as pd

nfa = {}

n = int(input("No. of states : "))

t = int(input("No. of transitions : "))

for i in range(n):

    state = input("state name : ")

    nfa[state] = {}

    for j in range(t):

        path = input("path : ")

        print("Enter end state from state {} travelling through path {} : ".format(state, path))

        reaching\_state = [x for x in input().split()]

        nfa[state][path] = reaching\_state

print("\nNFA :- \n")

print(nfa)

print("\nPrinting NFA table :- ")

nfa\_table = pd.DataFrame(nfa)

print(nfa\_table.transpose())

print("Enter final state of NFA : ")

nfa\_final\_state = [x for x in input().split()]

new\_states\_list = []

#-------------------------------------------------

dfa = {}

keys\_list = list(

    list(nfa.keys())[0])

path\_list = list(nfa[keys\_list[0]].keys())

dfa[keys\_list[0]] = {}

for y in range(t):

    var = "".join(nfa[keys\_list[0]][

                      path\_list[y]])

    dfa[keys\_list[0]][path\_list[y]] = var

    if var not in keys\_list:

        new\_states\_list.append(var)

        keys\_list.append(var)

while len(new\_states\_list) != 0:

    dfa[new\_states\_list[0]] = {}

    for \_ in range(len(new\_states\_list[0])):

        for i in range(len(path\_list)):

            temp = []

            for j in range(len(new\_states\_list[0])):

                temp += nfa[new\_states\_list[0][j]][path\_list[i]]

            s = ""

            s = s.join(temp)

            if s not in keys\_list:

                new\_states\_list.append(s)

                keys\_list.append(s)

            dfa[new\_states\_list[0]][path\_list[i]] = s

    new\_states\_list.remove(new\_states\_list[0])

print("\nDFA :- \n")

print(dfa)

print("\nPrinting DFA table :- ")

dfa\_table = pd.DataFrame(dfa)

print(dfa\_table.transpose())

dfa\_states\_list = list(dfa.keys())

dfa\_final\_states = []

for x in dfa\_states\_list:

    for i in x:

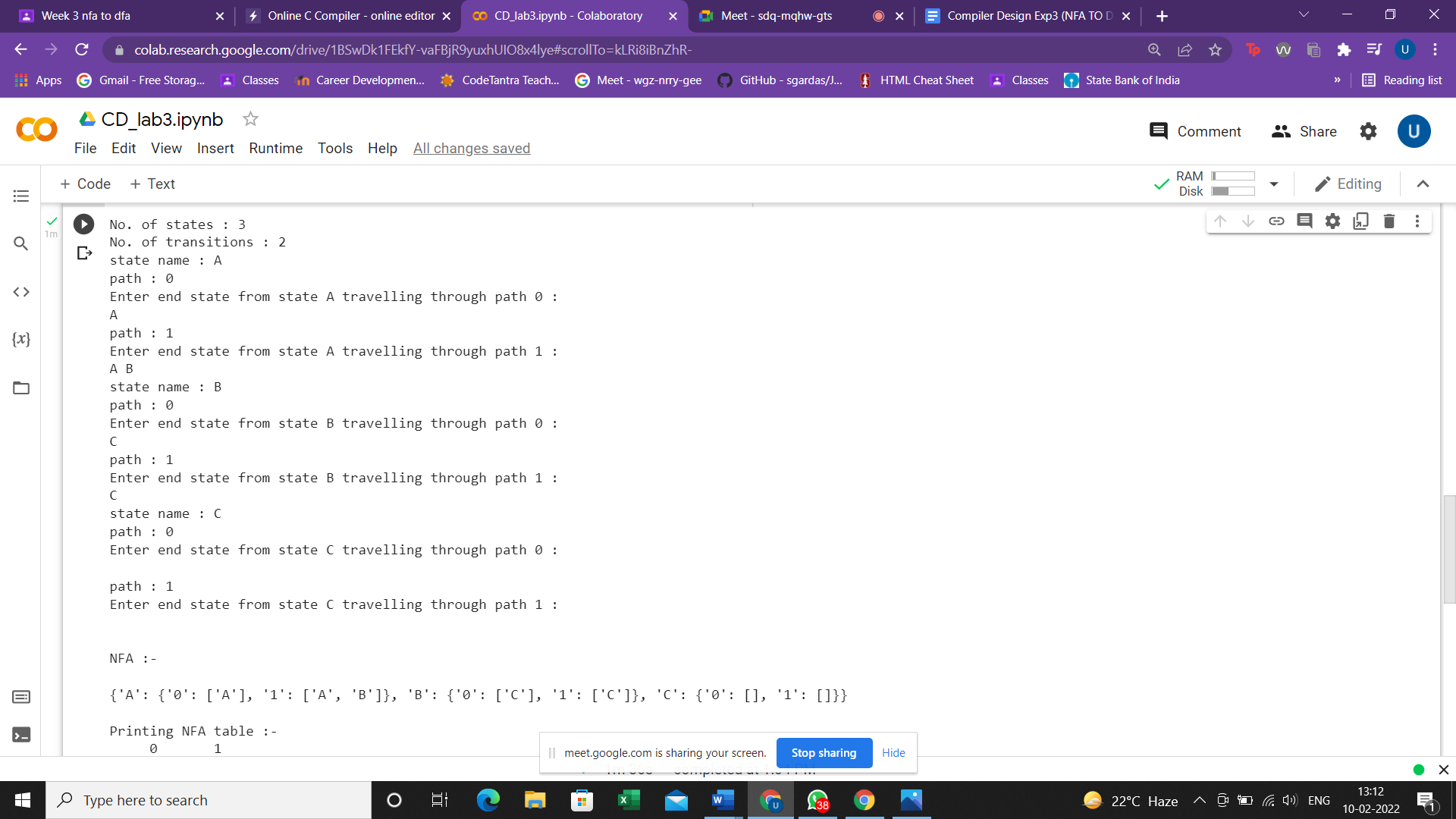
        if i in nfa\_final\_state:

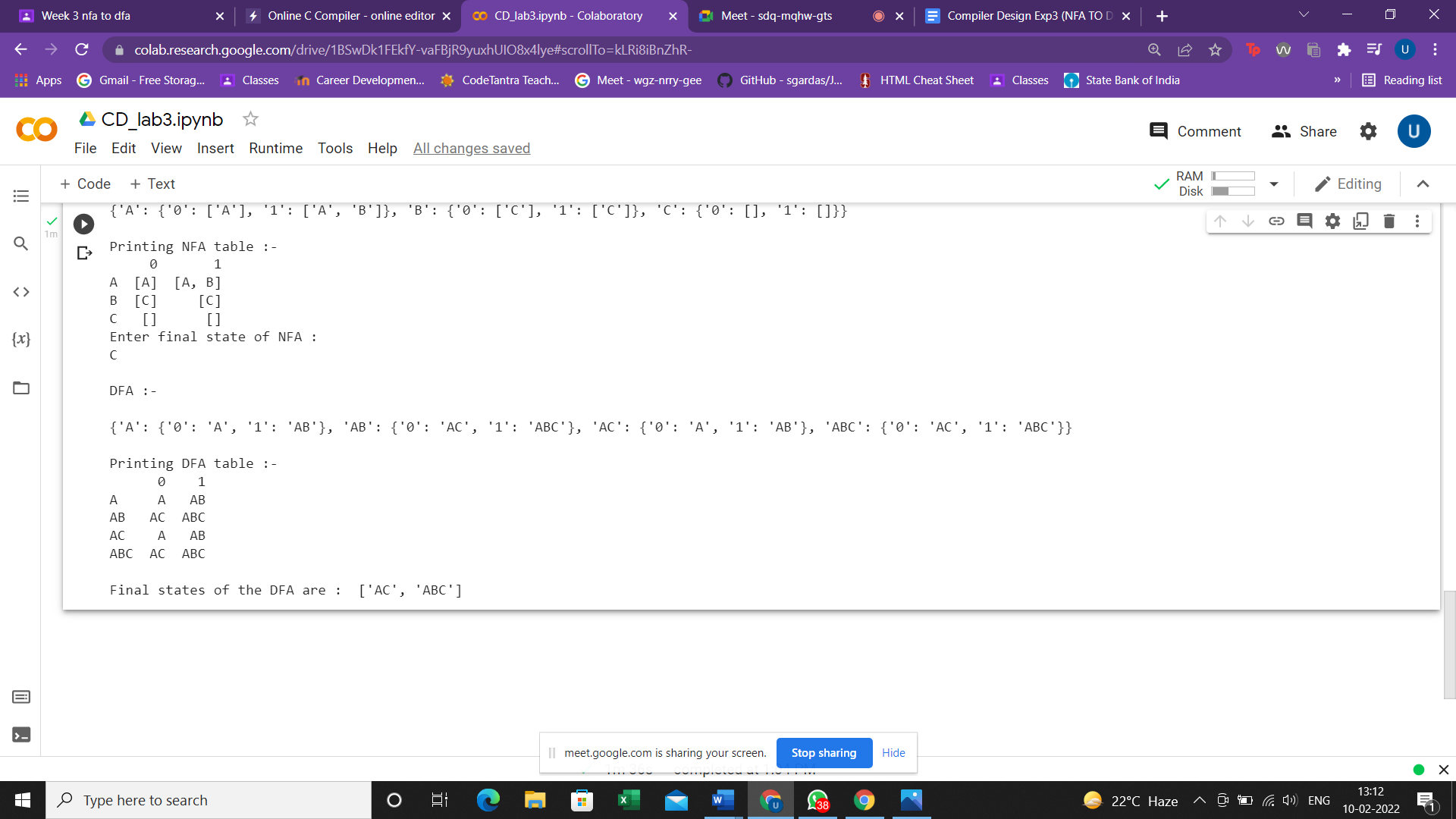
            dfa\_final\_states.append(x)

            break

print("\nFinal states of the DFA are : ", dfa\_final\_states)

**OUTPUT:**





**RESULT:**

The program to convert NFA into DFA in C language was successfully implemented.