**2: Brute Force Method**

**AASTHA SOOD**

**16BCE1104**

**Q1 CRYPTARITHM**

Consider the equation APPLE + LEMON = BANANA. Assume that each letter actually represents a digit from 0 to 9. Some conditions are imposed. The leftmost letter can't be zero in any word. There must be a one-to-one mapping between letters and digits. In other words, if you choose the digit 5 for the letter E, then all of the E's in the equation must be 5 and no other letter can be a 5. No digit can be repeated. What algorithmic strategy will you use for solving this problem? Write a Python program to solve the above equation.

**STRATEGY USED:**

Brute force method is used to find the values of the letters which satisfies the equation. Multiple loops and conditional IF statements are used to find all possible values.

Python Code:

for a in range(1,9,1):

for p in range(1,10,1):

for l in range(1,10,1):

for e in range(0,10,1):

for m in range(0,10,1):

for o in range(0,10,1):

for n in range(0,10,1):

for b in range(1,10,1):

if a!=p and a!=l and a!=e and a!=m and a!=o and a!=n and a!=b and p!=l and p!=e and p!=m and p!=o and p!=n and p!=b and l!=e and l!=m and l!=o and l!=n and l!=b and m!=o and m!=n and m!=b and o!=n and o!=b and n!=b and o!=b:

apple = a\*10000+p\*1000+p\*100+l\*10+e

lemon = l\*10000+e\*1000+m\*100+o\*10+n

banana = b\*100000+a\*10000+n\*1000+a\*100+n\*10+a

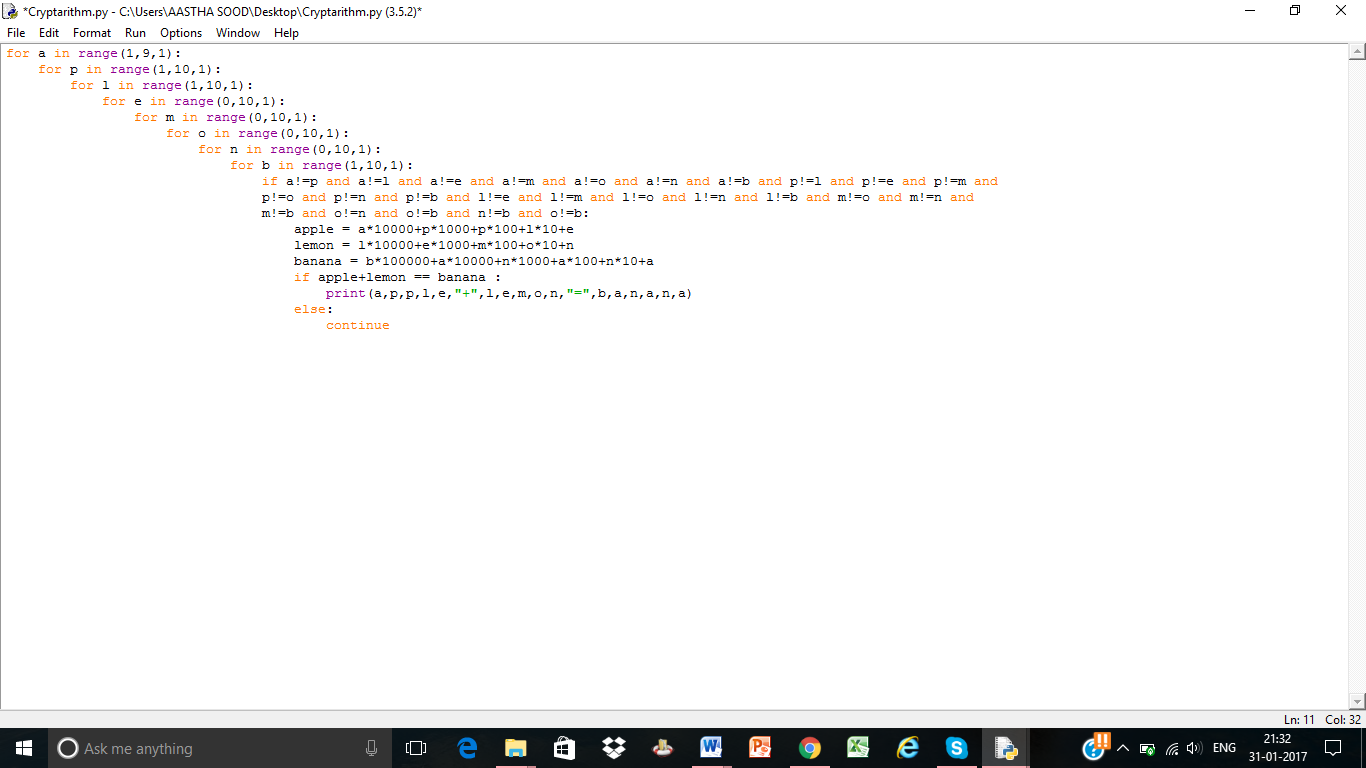
if apple+lemon == banana :

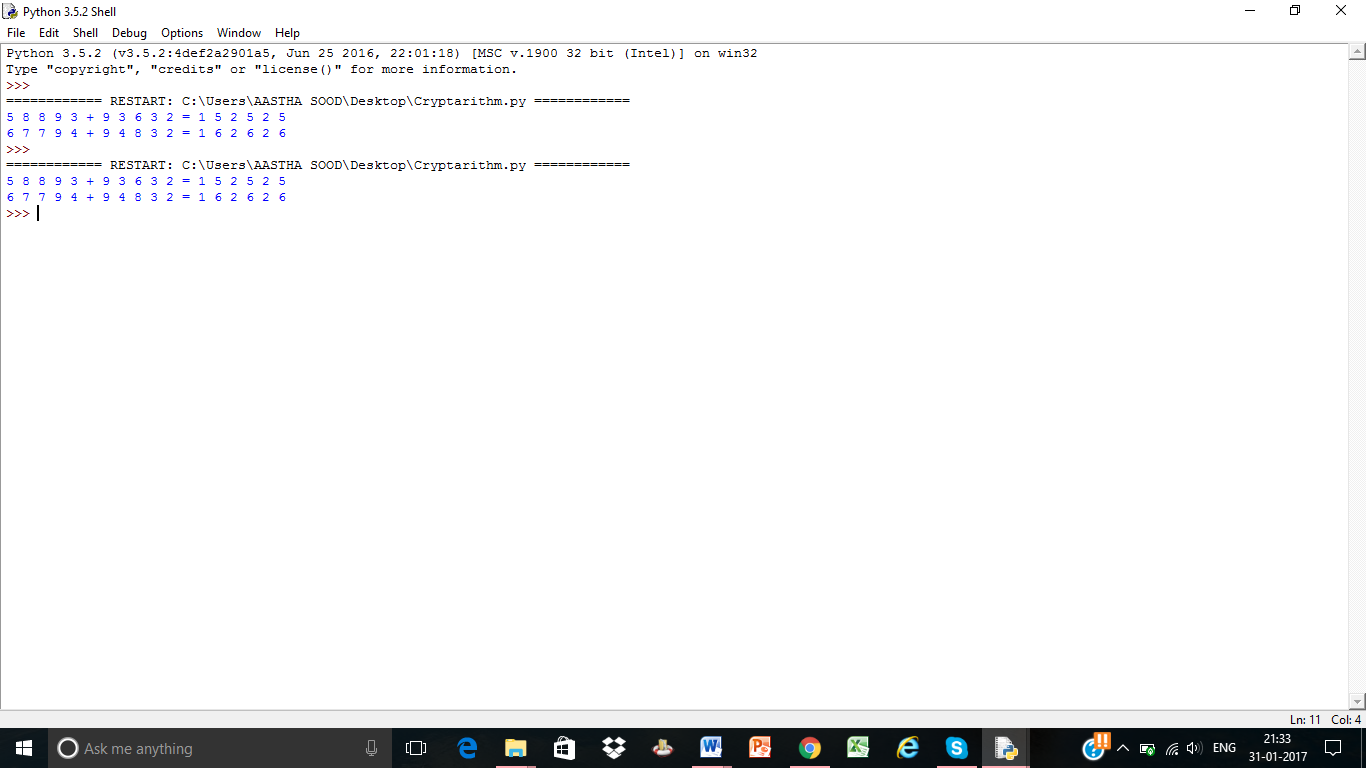
print(a,p,p,l,e,"+",l,e,m,o,n,"=",b,a,n,a,n,a)

else:

continue

Program and output screenshot:





Q2. **TRAVELLING SALESMAN PROBLEM**

The travelling salesman problem (TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" What algorithmic strategy will you use for solving this problem? What is the time complexity of your algorithm? Implement your code using Python.

**STRATEGY USED:**

In this question, the salesman can visit each city only once. So the distance between all the cities will be calculated. Here, we will use the Brute Force method as this method gives the most optimal solution to this problem.

**PYTHON CODE:**

def Dist(i,j):

if (i==1 and j==2 )or (i==2 and j==1):

return(cities[0])

if (i==1 and j==3) or (i==3 and j==1):

return(cities[1])

if (i==1 and j==4) or (i==4 and j==1):

return(cities[2])

if (i==2 and j==3) or (i==3 and j==2):

return(cities[3])

if (i==2 and j==4) or (i==4 and j==2):

return(cities[4])

if (i==3 and j==4) or (i==4 and j==3):

return(cities[5])

cities=[]

temp=9999999;d=0

for i in range(1,5):

for j in range(i+1,5):

print("Enter distance between city ",i," and ",j);

cities.append(int(input()))

for a in range(1,5):

for b in range(1,5):

for c in range(1,5):

for d in range(1,5):

L=[a,b,c,d]

if (len(set(L)))==4:

d=Dist(a,b)+Dist(b,c)+Dist(c,d)+Dist(d,a)

if d<temp:

temp=d

A,B,C,D=a,b,c,d

print(A,B,C,D,A)

print(temp)

Program and output screenshot:

