**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE CODE:** DJ19ITL406

**COURSE NAME:** Programing Laboratory 2 (Python)  **CLASS:** SYBTECH

**EXPERIMENT NO. 4**

**CO/LO: CO1, CO2.**

**AIM / OBJECTIVE:**

Write python programs to demonstrate applications of different decision-making statements

**DESCRIPTION OF EXPERIMENT:**

Python Conditions and **If** statements

Python supports the usual logical conditions from mathematics:

* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b

These conditions can be used in several ways, most commonly in "if statements" and loops.An "if statement" is written by using the if keyword.

## **Elif**

The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

## **Else**

The else keyword catches anything which isn't caught by the preceding conditions.

## **Nested If**

You can have if statements inside if statements, this is called *nested* if statements.

## **while Loop**

With the while loop we can execute a set of statements as long as a condition is true.

## **break Statement**

With the break statement we can stop the loop even if the while condition is true:

## **continue Statement**

With the continue statement we can stop the current iteration, and continue with the next:

## **For Loops**

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

## **range() Function**

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

**QUESTIONS:**

1. WAP to Reverse a Number

num=int(input("enter a number"))

temp=num

rev=0

while(temp>0):

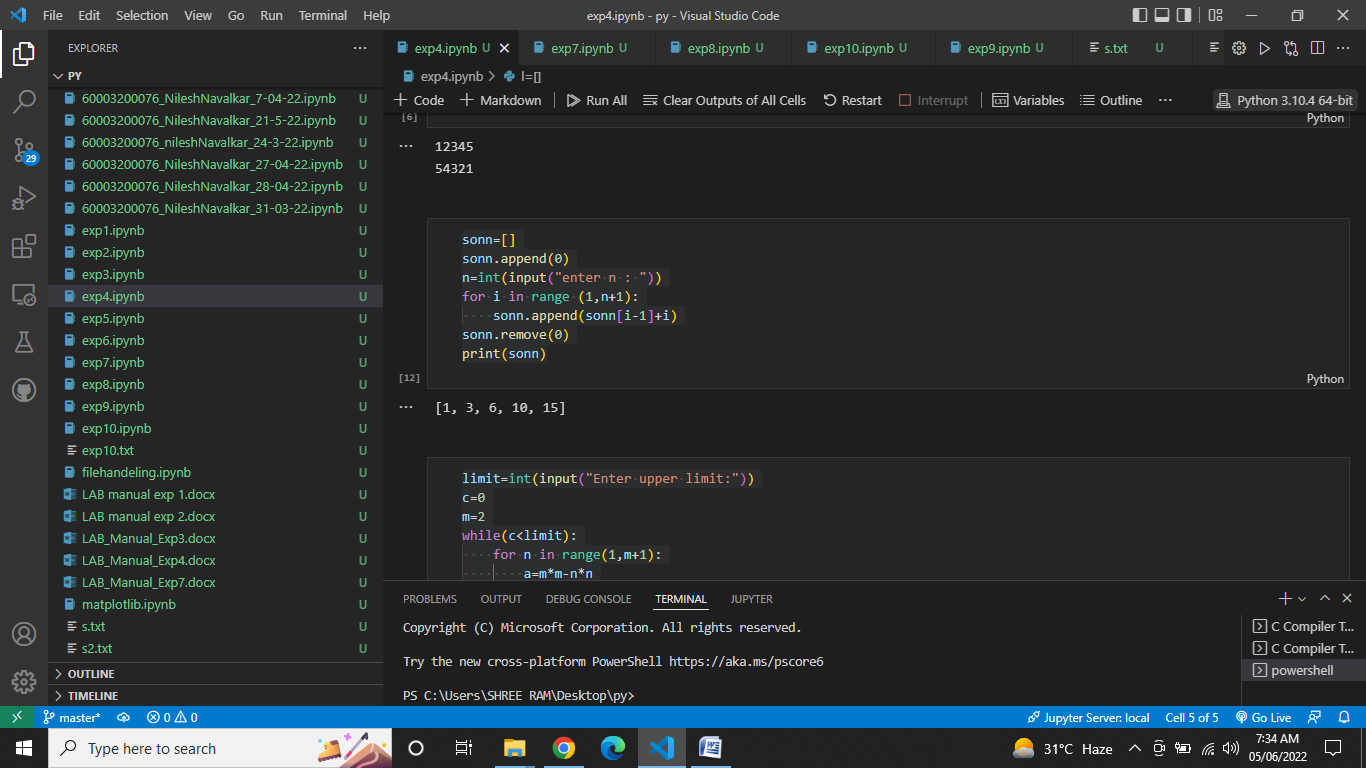
    d=temp%10

    rev=10\*rev+d

    temp=temp//10

print(num)

print(rev)



1. WAP to read number N and print natural numbers summation pattern

sonn=[]

sonn.append(0)

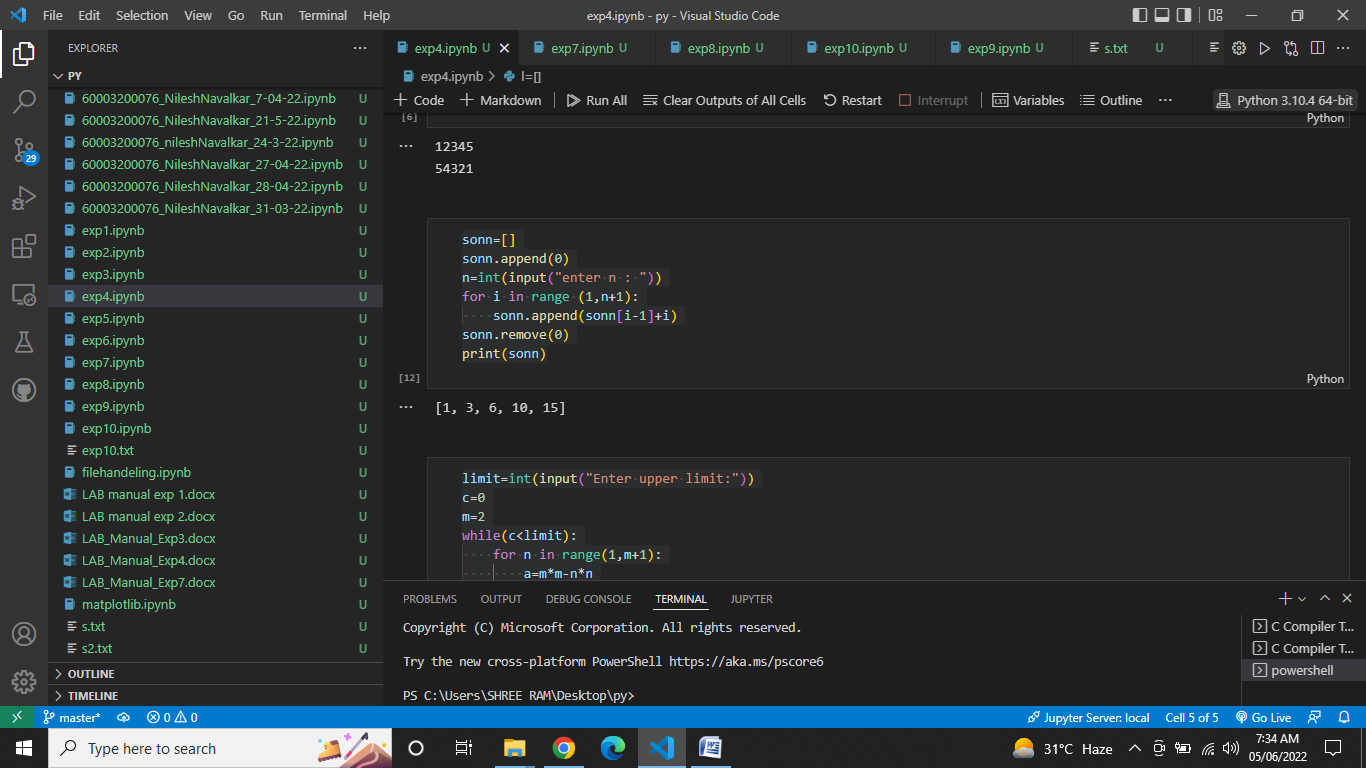
n=int(input("enter n : "))

for i in range (1,n+1):

    sonn.append(sonn[i-1]+i)

sonn.remove(0)

print(sonn)



1. WAP to determine all Pythagorean triplets

limit=int(input("Enter upper limit:"))

c=0

m=2

while(c<limit):

    for n in range(1,m+1):

        a=m\*m-n\*n

        b=2\*m\*n

        c=m\*m+n\*n

        if(c>limit):

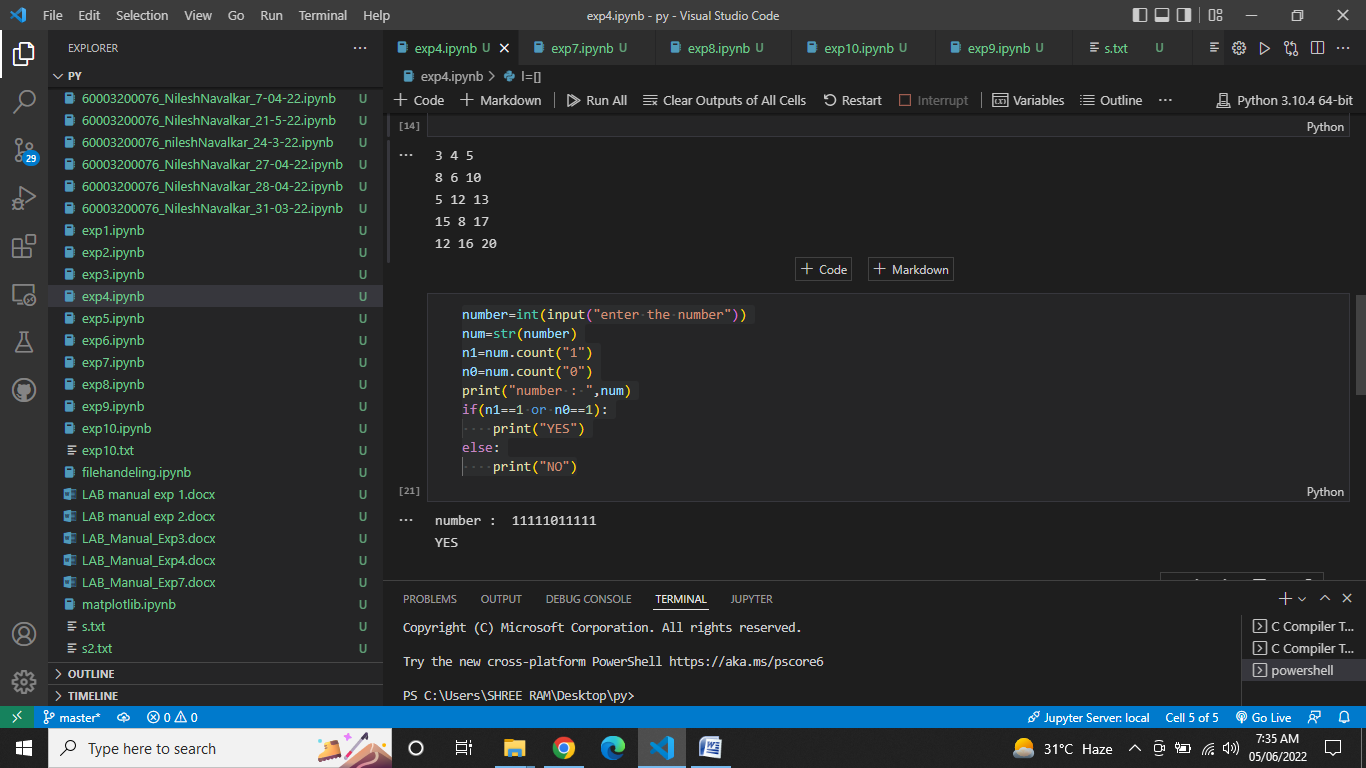
            break

        if(a==0 or b==0 or c==0):

            break

        print(a,b,c)

    m=m+1



1. WAP, You are given a number A which contains only digits 0's and 1's. Your task is to make all digits same by just flipping one digit (i.e. 0 to 1 or 1 to 0 ) only. If it is possible to make all the digits same by just flipping one digit then print 'YES' else print 'NO'.

number=int(input("enter the number"))

num=str(number)

n1=num.count("1")

n0=num.count("0")

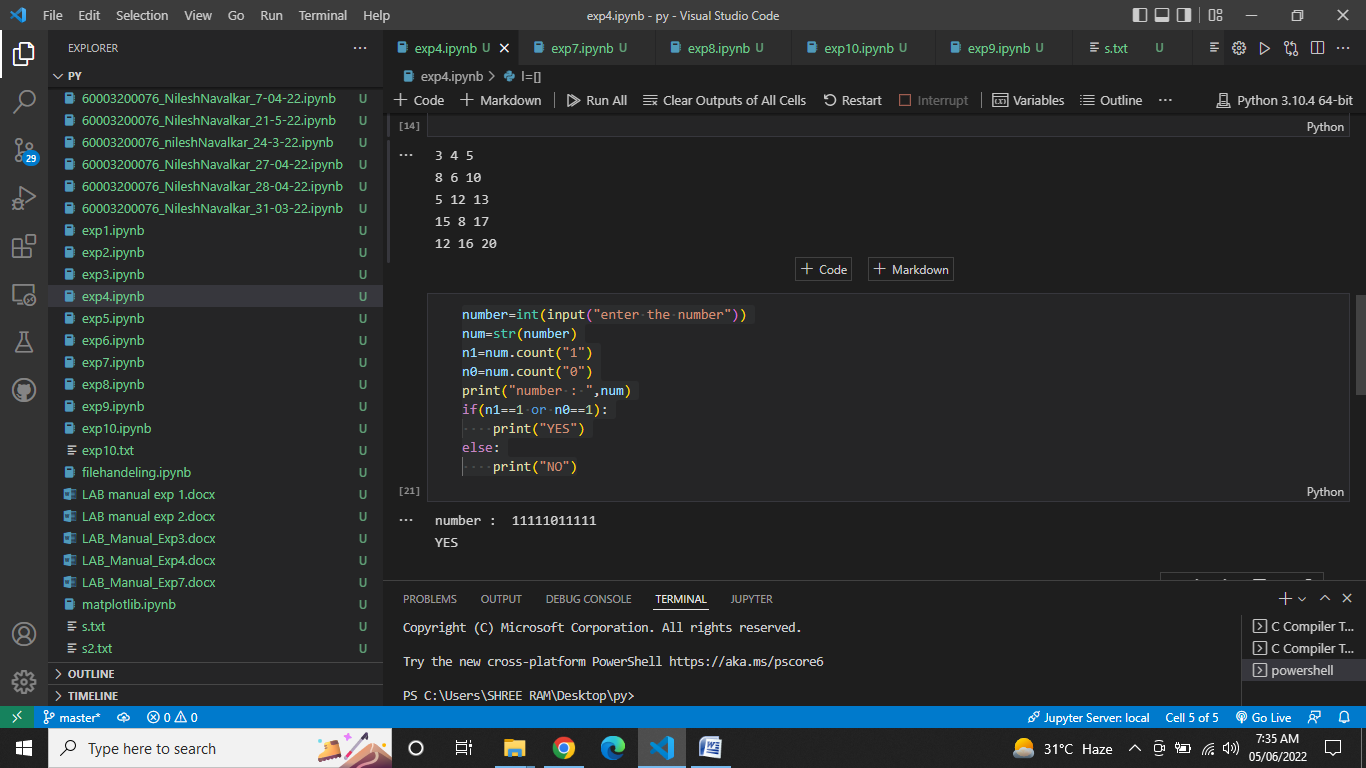
print("number : ",num)

if(n1==1 or n0==1):

    print("YES")

else:

    print("NO")



1. WAP, Given a list A of N distinct integer numbers, you can sort the list by moving an element to the end of the list. Find the minimum number of moves required to sort the list using this method in ascending order.

l=[]

n=int(input("enter n"))

for i in range(0,n):

    e=int(input())

    l.append(e)

print(l)

j=n

count=0

while(j>=1):

    m=max(l[0:j])

    indx=l.index(m)

    if(indx!=j-1):

        temp=l[indx]

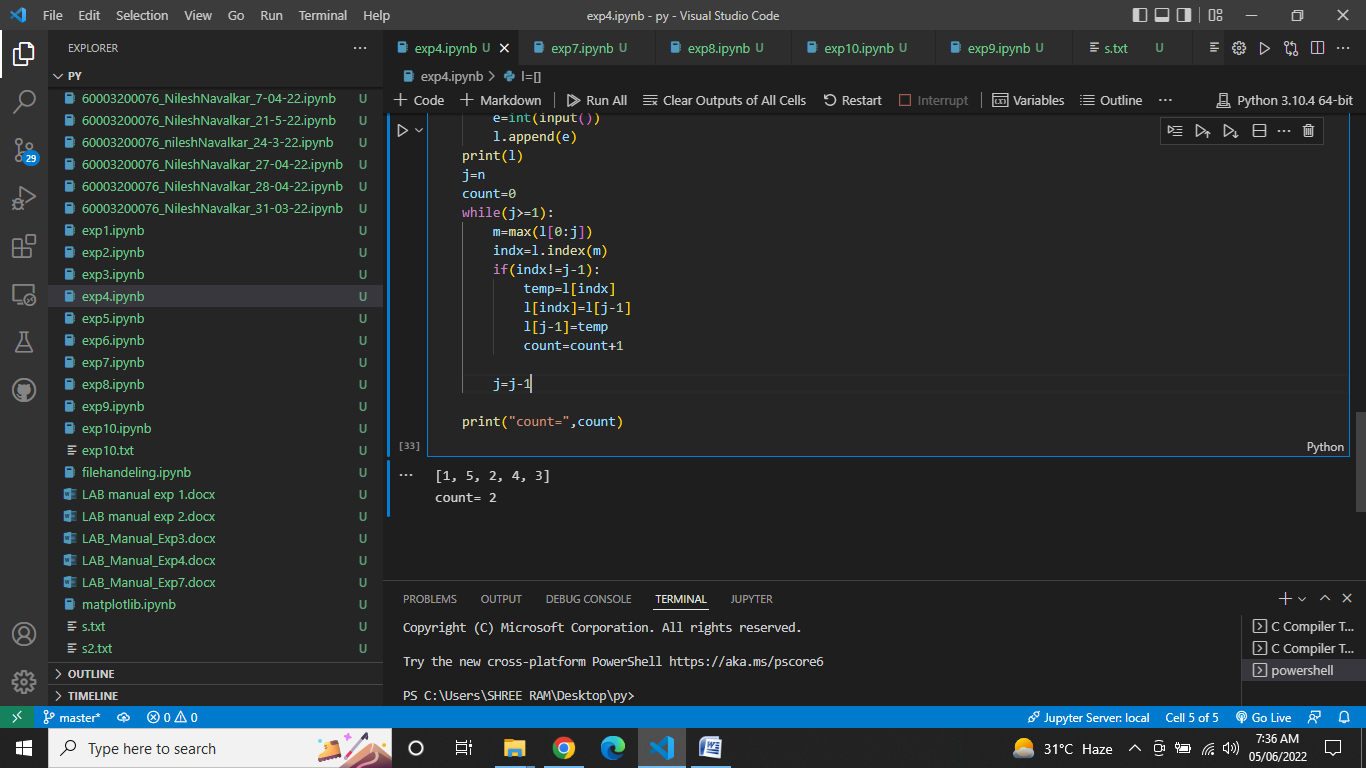
        l[indx]=l[j-1]

        l[j-1]=temp

        count=count+1

    j=j-1

print("count=",count)

****

**OBSERVATIONS / DISCUSSION OF RESULT:**

**CONCLUSION:**

**REFERENCES:**

**Website References:​**

[1] https://www.w3schools.com/python