NAME : KOVI SAI GANESH

REGISTER NUMBER : 192124028

DAY 5

10q Reverse Words in a String Given an input string s, reverse the order of the words. A word is defined as a sequence of non-space characters. The words in s will be separated by at least one space. Return a string of the words in reverse order concatenated by a single space. Note that s may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces. Test Cases: 1.Input: s = "the sky is blue" Output: "blue is sky the" 2.Input: s = " hello world " Output: "world hello" 3.Input: s = "a good example" Output: "example good a" 4. s= “apple is red” 5.s= “Red rose”

def reverse\_word(s, start, end):

while start < end:

s[start], s[end] = s[end], s[start]

start = start + 1

end -= 1

s = input("Enter Your String : " )

s = list(s)

start = 0

while True:

try:

end = s.index(' ', start)

reverse\_word(s, start, end - 1)

start = end + 1

except ValueError:

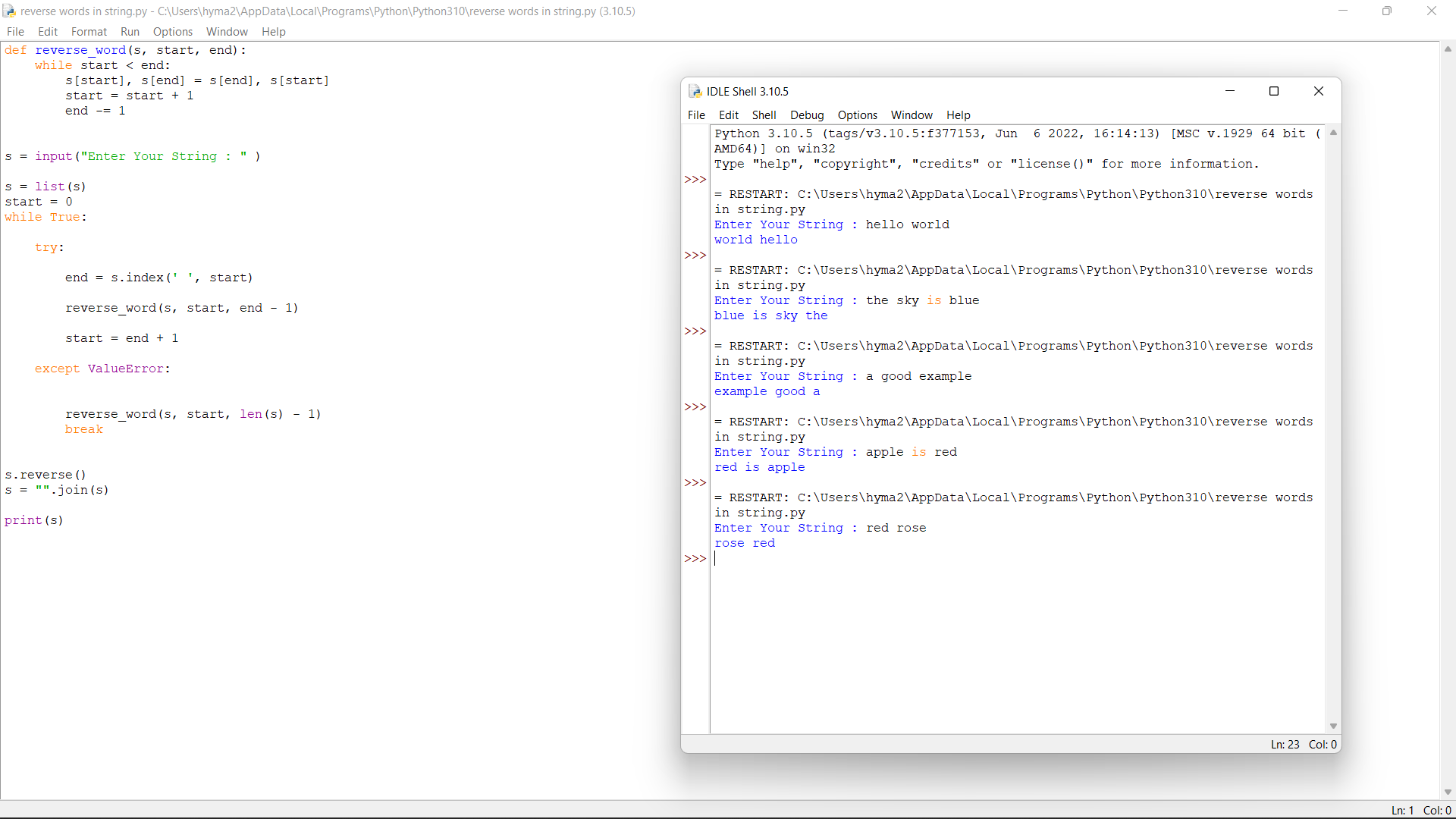
reverse\_word(s, start, len(s) - 1)

break

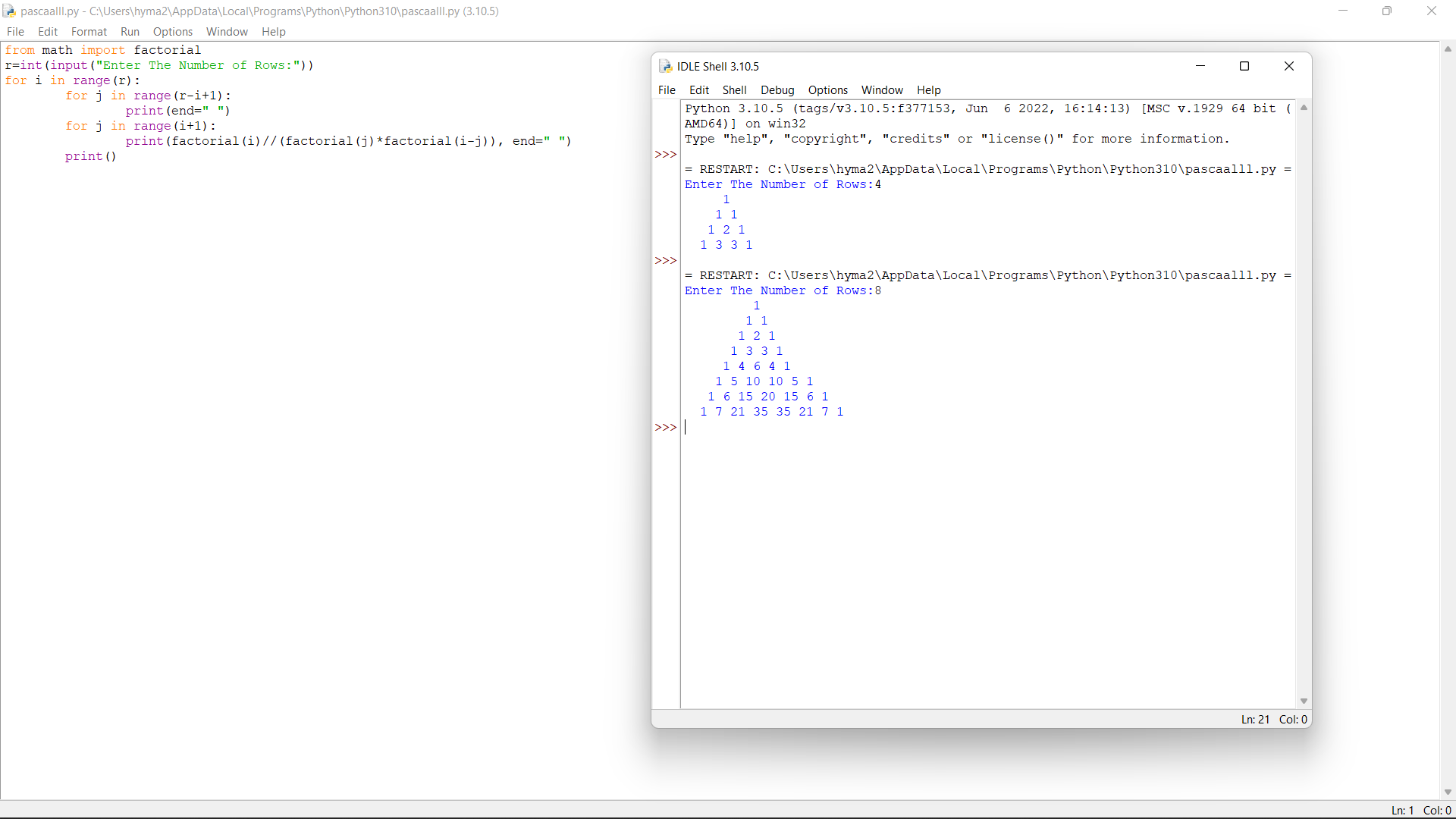
s.reverse()

s = "".join(s)

print(s)



6. Implement a triangular array of the binomial coefficients that arises in probability theory, combinatorics, and algebra. Find the sum of elements in the nth row. Sample Input: Enter the number of rows: 5 Enter the row number: 4 Sample Output: Sum of elements in 4th row: 8



2. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Sample Input & Output:

Enter the grade of the employee: B

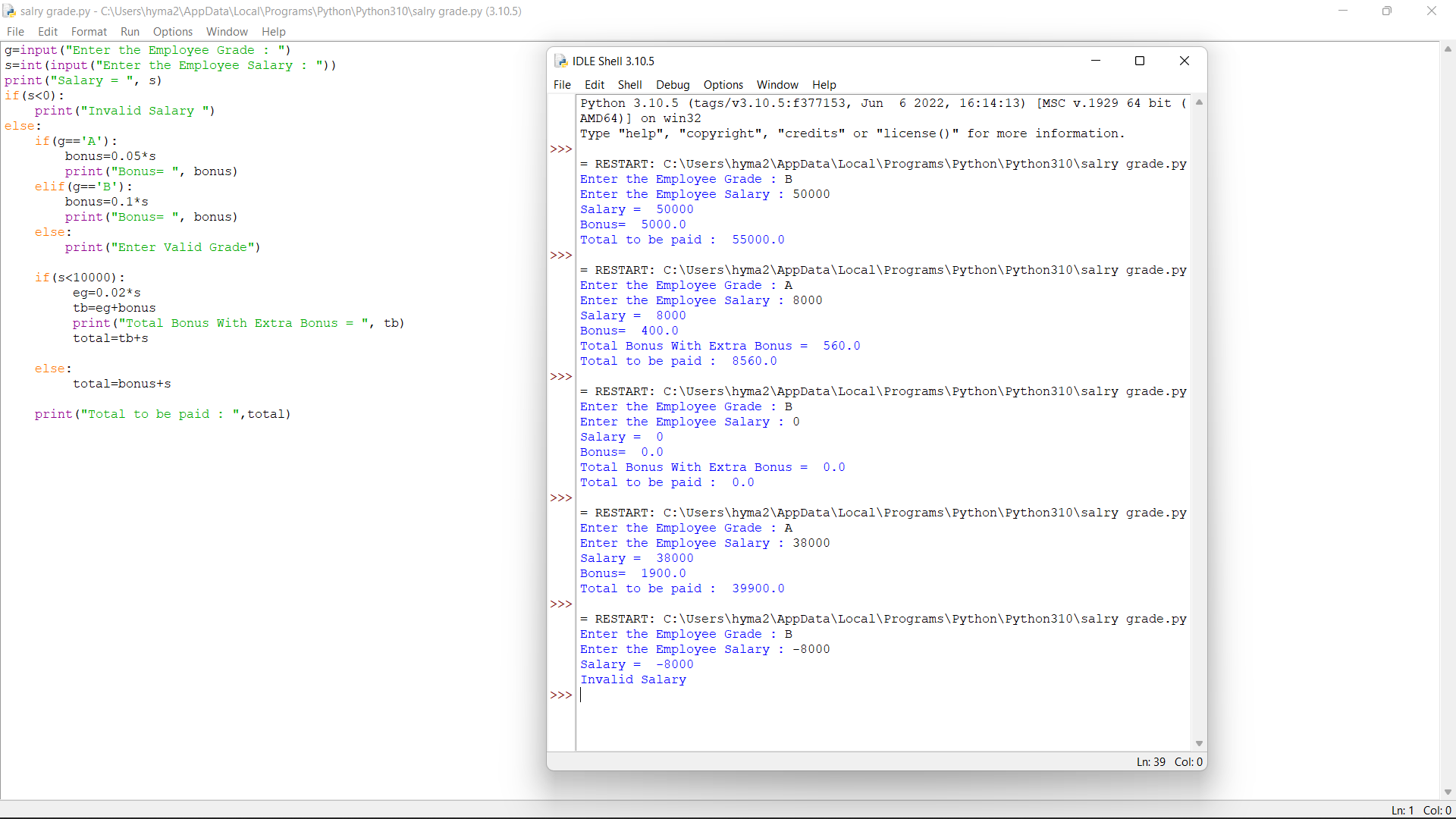
Enter the employee salary: 50000

Salary=50000 Bonus=5000.0

Total to be paid:55000.0

Test cases:

1. Enter the grade of the employee: A Enter the employee salary: 8000 2. Enter the grade of the employee: C Enter the employee salary: 60000 3. Enter the grade of the employee: B Enter the employee salary: 0 4. Enter the grade of the employee: 38000 Enter the employee salary: A 5. Enter the grade of the employee: B Enter the employee salary: -8000



5. A peak element is an element that is strictly greater than its neighbors.Given a 0- indexed integer array nums, find a peak element, and return its index. If the array contains multiple peaks, return the index to any of the peaks.You may imagine that nums[-1] = nums[n] = -∞. In other words, an element is always considered to be strictly greater than a neighbor that is outside the array.You must write an algorithm that runs in O(log n) time.

Test Case: 1.Input: nums = [1,2,3,1] Output: 2 2.Input: nums = [1,2,1,3,5,6,4] Output: 5 3.Input: nums = [5, 10, 20, 15] Output: 20 4.Input: nums = [ 1, 3, 2, 4, 1, 0 ] Output: 4 5.Input: nums = [1,1,1,3,2,1,2] Output: 3

def findPeak(arr, n) :

# first or last element is peak element

if (n == 1) :

return 0

if (arr[0] >= arr[1]) :

return 0

if (arr[n - 1] >= arr[n - 2]) :

return n - 1

# check for every other element

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

# check if the neighbors are smaller

if (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]) :

return i

# Driver code.

arr = eval(input("Enter The List : "))

n = len(arr)

print("Index of a peak point is", findPeak(arr, n))

9. Write a function shuffle(l1,l2) that takes as input two lists, 11 and l2, and returns a list consisting of the first element in l1, then the first element in l2, then the second element in l2, then the second element in l2, and so on. If the two lists are not of equal length, the remaining elements of the longer list are appended at the end of the shuffled output. Sample Input: Enter the number of elements of l1: 3 Enter the element: 1 Enter the element: 3 Enter the element: 5 Enter the number of elements of l2:5 Enter the element: 2 Enter the element: 4 Enter the element: 6 Enter the element: 8 Enter the element: 10 Sample output: Shuffled list=[1,2,3,4,5,6,8,10] Test cases: 1. 0,4 2. -1,1 3. 5,5 4. 10,5 5. 6,9

def maxaverage(l):

d = {}

for i in l:

name, score = i

if name in d:

tot\_score, num = d[name]

d[name] = (tot\_score+score, num+1)

else:

d[name] = (score, 1)

max= -1

for key in d:

tot\_score, num = d[key]

ave = tot\_score/num

if(max < ave):

max = ave

l = []

for key in d:

tot\_score, num = d[key]

ave = tot\_score/num

if(max == ave):

l.append(key)

l.sort()

return l

3. Write an algorithm to determine if a number n is happy. A happy number is a number defined by the following process: • Starting with any positive integer, replace the number by the sum of the squares of its digits. • Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. • Those numbers for which this process ends in 1 are happy. Return true if n is a happy number, and false if not. Test Case 1. Input: n = 19 Output: true Explanation: 12 + 92 = 82 82 + 22 = 68 62 + 82 = 100 12 + 02 + 02 = 1 2. Input: n = 2 Output: false 3. Input: n=-1 4. Input: n=0 5. 5

def numSquareSum(n):

squareSum = 0

while(n):

squareSum += (n%10)\*(n%10)

n=int(n/10)

return squareSum

def isHappynumber(n):

slow=n

fast=n

while(True):

slow=numSquareSum(slow)

fast=numSquareSum(numSquareSum(fast))

if(slow!=fast):

continue

else:

break

return (slow==1)

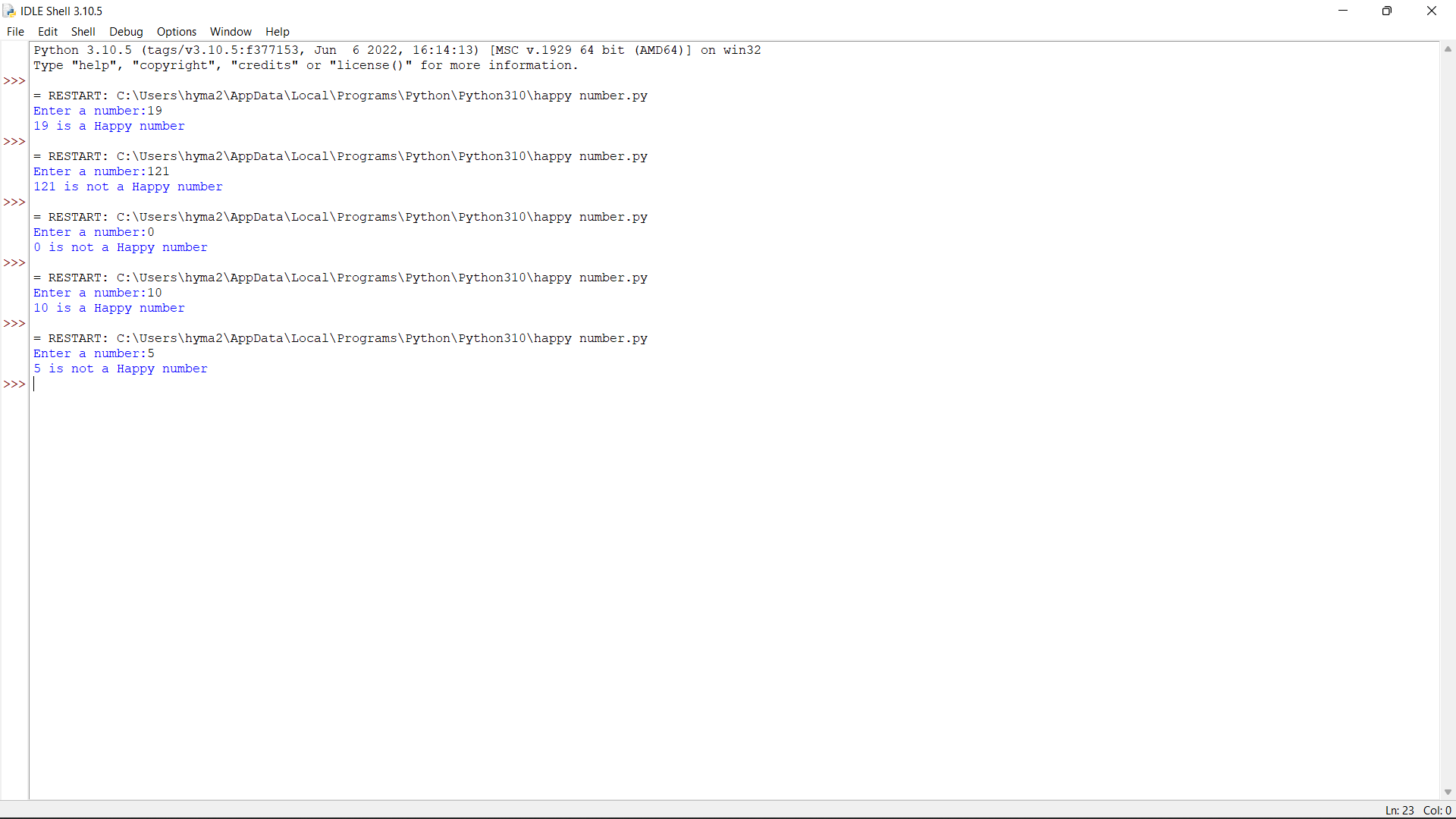
n =int(input("Enter a number:"))

if (isHappynumber(n)):

print(n , "is a Happy number")

else:

print(n , "is not a Happy number")



4. Given an integer x, return true if x is palindrome integer. An integer is a palindrome when it reads the same backward as forward. For example, 121 is a palindrome while 123 is not. Test cases: 1.Input: x = 121 Output: true Explanation: 121 reads as 121 from left to right and from right to left. 2.Input: x = -121 Output: false Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome. 3.Input: x = 10 Output: false Explanation: Reads 01 from right to left. Therefore it is not a palindrome. 4.x=abc 5.x=0

num=int(input("Enter a number:"))

temp=num

rev=0

while(num>0):

dig=num%10

rev=rev\*10+dig

num=num//10

if(temp==rev):

print("The number is palindrome!")

else:

print("Not a palindrome!")

